US Army Corps
of Engineers
Waterways Experiment
Station

Instruction Report W-93-2
August 1993

2

Water Operations Technical Support Program

WESTEX: A Numerical, One-Dimensional Reservoir Thermal Model

Report 2
Programmer's Manual, Version 3.0

Edited by *Darrell G. Fontane*
Colorado State University

Stacy E. Howington, Michael L. Schneider,
Steven C. Wilhelms
Hydraulics Laboratory

Approved For Public Release; Distribution Is Unlimited

Prepared for Headquarters, U.S. Army Corps of Engineers

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

**Water Operations Technical
Support Program**

Instruction Report W-93-2
August 1993

WESTEX: A Numerical, One-Dimensional Reservoir Thermal Model

Report 2
Programmer's Manual, Version 3.0

Edited by Darrell G. Fontane

Department of Civil Engineering
Colorado State University
Fort Collins, CO 80523

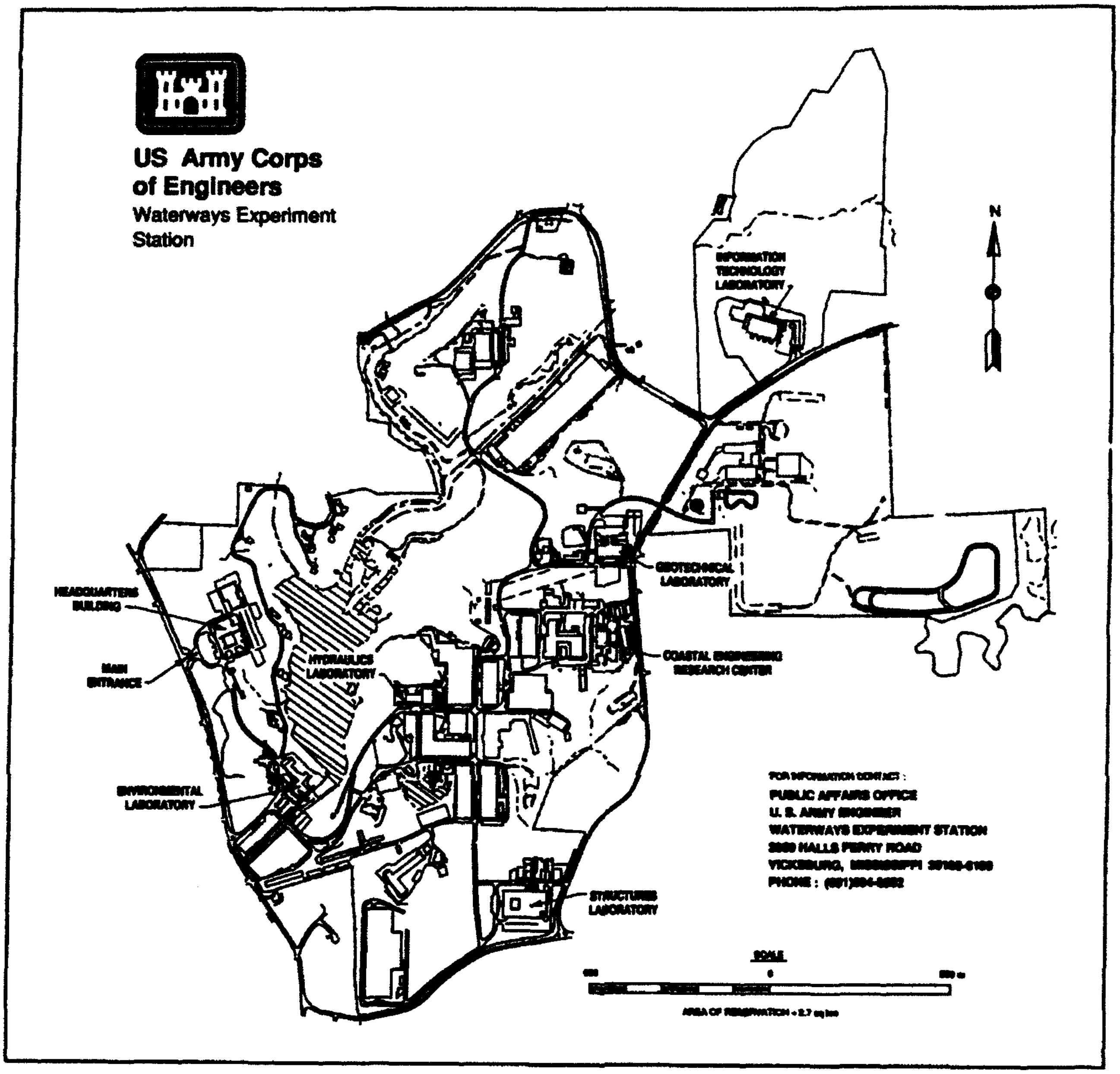
Stacy E. Howington, Michael L. Schneider,
Steven C. Wilhelms

Hydraulics Laboratory

U.S. Army Corps of Engineers
Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Report 2 of a series

Prepared for Headquarters, U.S. Army Corps of Engineers
Washington, DC 20314-1000



Waterways Experiment Station Cataloging-in-Publication Data

Fontane, Darrell G.
WESTEX : a numerical, one-dimensional reservoir thermal model. Report 2, Programmer's manual, version 3.0 / edited by Darrell G. Fontane, Stacy E. Howington, Michael L. Schneider ; prepared for Headquarters, U.S. Army Corps of Engineers.
122 p. : ill. ; 28 cm. — (Instruction report ; W-93-2 rept. 2)
1. Water temperature — Mathematical models. 2. Reservoir drawdown — Management — Mathematical models. 3. Water quality management — Mathematical models. I. Fontane, Darrell G. II. Howington, Stacy E. III. Schneider, Michael L. IV. United States. Army. Corps of Engineers. V. U.S. Army Engineer Waterways Experiment Station. VI. Water Operations Technical Support Program. VII. Title: WESTEX: a numerical, one-dimensional reservoir thermal model. Report 2, Programmer's manual, version 3.0. VIII. Series: Instruction report (U.S. Army Engineer Waterways Experiment Station) ; W-93-2 rept. 2.
TA7 W34i no.W-93-2 rept.2

PREFACE

The work reported herein was conducted as part of the Water Operations Technical Support (WOTS) program. The WOTS is sponsored by the Headquarters, US Army Corps of Engineers (HQUSACE), and is assigned to the US Army Engineer Waterways Experiment Station (WES) under the purview of the Environmental Laboratory (EL). Funding was provided under Department of the Army Appropriation 96X3123, Operations and Maintenance. The WOTS is managed under the Environmental Resources Research and Assistance Programs (ERRAP), Mr. J. L. Decell, Manager. Dr. A. J. Anderson was Assistant Manager, ERRAP, for the WOTS. Technical Monitors during this study were Messrs. F. B. "Pete" Juhle and Jim Gottesman, HQUSACE.

This programmer's guide was prepared as a technology transfer activity. This is Report 2 of a series. Report 1 is a user's guide. This guide was compiled from information contained in technical reports and papers previously developed at WES. The information was organized, prepared, and edited by Dr. Darrell G. Fontane, Department of Civil Engineering, Colorado State University, Fort Collins, CO, under the Inter-Governmental Personnel Agreement, and Messrs. Stacy E. Howington, Michael L. Schneider, and Steven C. Wilhelms of the Reservoir Water Quality Branch (RWQB), Hydraulic Structures Division (HSD), Hydraulics Laboratory (HL), WES. Dr. Jeffery P. Holland, Director, Computational Hydraulics Institute, HL, contributed significantly to the overall organization of the user's manual. This report is based to a large extent upon the draft of an earlier user's manual developed by Dr. Bruce Loftis, formerly of the RWQB. The report was prepared under the general supervision of Messrs. F. A. Herrmann, Jr., Director, HL; R. A. Sager, Assistant Director, HL; and G. A. Pickering, Chief, HSD, HL.

At the time of publication of this report, Director of WES was Dr. Robert W. Whalin. Commander was COL Leonard G. Hassell, EN.

This report should be cited as follows:

Fontane, D. G., Howington, S. E., Schneider, M. L., and Wilhelms, S. C. 1993. "WESTEX: A Numerical, One-Dimensional Reservoir Thermal Model; Report 2, Programmer's Manual, Version 3.0", Instruction Report W-93-2, US Army Engineer Waterways Experiment Station, Vicksburg, MS.

CONTENTS

Page

WESTEX: A NUMERICAL, ONE-DIMENSIONAL RESERVOIR THERMAL MODEL
PROGRAMMER'S MANUAL, VERSION 3.0

Summary

1. WESTEX is a numerical, one-dimensional reservoir model that predicts the thermal stratification pattern and the distribution of conservative water quality parameters. This report is a companion document to the WESTEX User's Manual.* The WESTEX User's Manual describes the fundamental concepts of thermal stratification processes in reservoirs, describes the components of the WESTEX model and its computer implementation as Version 3.0, and includes examples of model input and output. The purpose of the Programmer's Manual is to supplement the User's Manual by providing a complete listing of the model's source code for Version 3.0 and a definition of the FORTRAN variables used in the code.

2. This report is organized into two appendices. Appendix A contains the source code listing of the program organized by subprograms. The subprograms are arranged in alphabetical order. Appendix B contains the definitions of variables used in the code also arranged by subprogram.

3. Inquiries concerning the WESTEX model may be directed to Michael L. Schneider, Reservoir Water Quality Branch (Physical), Hydraulics Laboratory, US Army Engineer Waterways Experiment Station, at (601) 634-3424.

* Fontane, D. G., Howington, S. E., Schneider, M. L., and Wilhelms, S. C. 1993 (July). "WESTEX: A Numerical, One-Dimensional Reservoir Thermal Model; Report 1, User's Manual," Instruction Report W-93- , US Army Engineer Waterways Experiment Station, Vicksburg, MS.

APPENDIX A: LISTING OF WESTEX COMPUTER CODE

```
         PROGRAM WESTEX
**************************************************************************
*       WESTEX  VERSION III    1992
*       CAN BE RUN IN EITHER VERIFICATION OR PREDICTION MODE
*       WITH PORTS OR WEIR ( VERIFICATION ONLY ) .
*       THIS IS A THERMAL MODEL FOR LAKES WITH CONSERVATIVE QUALITIES.
*       THE USER MUST SPECIFY THE QUALITY PARAMETERS AND
*       THEIR ASSOCIATED SOURCES, SINKS, AND REACTION
*       RATES.  QUALITIES CAN INFLUENCE DENSITY THROUGH
*       SPECIFICATION OF A DENSITY COEFFICIENT.
*
        COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
        COMMON / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
        COMMON / CC / DEN(100), NUSURF
        COMMON / CH / NM, TITLE
        COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
        COMMON / EE / KFILE, LFILE, JFILE, IFILE
        COMMON / FF / NPORTS, PAREA(8), PHGT(8)
        COMMON / GG / FLOWIN(3), TEMPIN(3), QUALIN(3, 3)
        COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
        COMMON / II / EK, ET, SHORT, BETA, LAMBDA, MIXCOEF, GAMMA,
     *                  DECAY
        COMMON / JJ / SUMOUT, WIDTH(100)
        COMMON / KK / NWELL(8), FMAX(8), FMIN(8), SELMAX
        COMMON / LL / FGAREA, FGHGT, FGMAX, FGMIN, FGANG, LPORT(8)
        COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
        COMMON / OO / AVTEMP, AVQUAL(3), AVGT(366), AVGQ(366, 3)
        COMMON / PP / FIRST, LAST, NJ, INDEX(366)
        COMMON / QQ / QVERI, QOUTC, QINTC, QINCFS, QOCFS, QWCFS
        COMMON / RR / INFLO(366, 3), INTEMP(366, 3), INQUAL(366, 3, 3)
        COMMON / SS / OUTFLO(366, 8), TARG(366)
        COMMON / TT / EQTEMP(366), EXCOEF(366), SOLAR(366)
        COMMON / UU / ENFLOW(100), SUMFLO(366), DPRINT(14)
        COMMON / VV / START, FINISH, QRELE, QPROF
        COMMON / WW / QNKWAL, YEAR(2)
        COMMON / XX / QJUNK, ONE, TWO
        COMMON / ZZ / DINIT, TINIT(100), QINIT(3, 100)
        COMMON / BC / WRLNG, WRGHT, WRTYPE, DCOEF
        COMMON / CD / QPORT, QWEIR, QVINCR, QINITC
        COMMON / DE / TOPLIM, LOWLIM, V(100), FLORAT, LL(8), LT(8)
        COMMON / EF / AREA, HGTPRT, LAYPRT, DENPRT
        COMMON / GH / WRFLOW, WRFLO(366)
        COMMON / BL / WIND(366)
        COMMON / IJ / WANGLE, WTHETA(8)
        CHARACTER TITLE*78, NM*3
        LOGICAL QVERI, QOUTC, QINTC, QFIRST, QPRINT,
     *          QINCFS, QOCFS, QJUNK, QNKWAL,
     *          QPORT, QWEIR, QVINCR, QRELE, QPROF,
     *          QWCFS, QINITC
        INTEGER PORT, OPEN, FIRST, YEAR, DPRINT,
     *          START, FINISH, ONE, TWO, DAY
        REAL INTEMP, INQUAL, INFLO, LAMBDA
        EQUIVALENCE ( DAY, N )
        NAMELIST / WNAME / DAY, DEPTH
        NAMELIST / WNAME1 / QUALIN, NIP, NQUAL
        DATA QFIRST / .TRUE. /
```

```
      DATA C1, C2 / - 3.9863, 508929.2 /
      DATA C3, C4 / 288.9414, 68.12963 /
      DENFUN ( T ) = 1. - ( T + C1 ) ** 2 / C2
     *               * ( T + C3 ) / ( T + C4 )
      OPEN(UNIT=5, FILE='WESTEX.IN',  STATUS='OLD')
      OPEN(UNIT=6, FILE='WESTEX.OUT', STATUS='UNKNOWN')
      OPEN(UNIT=10,FILE='WESTEX.PLT', STATUS='UNKNOWN')
 100  CONTINUE
      CALL XREAD
      CALL HMREAD
      CALL CONVRT
*
*  ESTABLISH INITIAL CONDITIONS
*
      DEPTH = DINIT
      ODEPTH = DEPTH
      LSURF = 1. + DEPTH / DELZ
      DO 120 I = 1, LSURF
      TEMP(I) = TINIT(I)
      IF ( QNKWAL ) GO TO 120
      DO 110 J = 1, NQUAL
      QUAL(J, I) = QINIT(J, I)
 110  CONTINUE
 120  CONTINUE
      IF ( .NOT. QFIRST ) GO TO 130
      QFIRST = .FALSE.
      QPRINT = DPRINT(1) .GT. 0
      IF ( QPRINT ) CALL XFIRST
      NFLOOD = NPORTS + 1
 130  CONTINUE
*
*  ESTABLISH THICKNESS OF EACH LAYER
*
      DO 140 I = 1, LSURF
      HGT(I) = 1.
 140  CONTINUE
      DO 150 I = LSURF, MAXLAY
      HGT(I) = 0.
 150  CONTINUE
      HGT (LSURF) = ( DEPTH - DELZ * FLOAT ( LSURF - 1 ) ) / DELZ
*
*  ESTABLISH INITIAL DENSITY PROFILE
*
      DO 180 I = 1, LSURF
      DENSQ = 0.
      IF ( QNKWAL ) GO TO 170
      DO 160 J = 1, NQUAL
      DENSQ = DENSQ + QUAL(J,I) * DENC(J)
 160  CONTINUE
 170  CONTINUE
      DENST = DENFUN ( TEMP(I) )
      DEN(I) = DENST + DENSQ
 180  CONTINUE
      AVTEMP = TEMP(1)
*
*  INITIATE LOOP OF DAYS
*
      DO 280 N = START, FINISH
      WRITE(*,'(A,I3)') '       SIMULATING DAY => ',N
      QJUNK = N .GE. ONE .AND. N .LE. TWO
*
*  PREPARE HEAT EXCHANGE DATA FOR
*  ONE DAY OF SIMULATION
*
         ET = EQTEMP(N)
         EK = EXCOEF(N)
         SHORT = SOLAR(N)
*
*  PREPARE INFLOW DATA FOR
*  ONE DAY OF SIMULATION
*
         DO 200 L = 1, NIP
         FLOWIN(L) = INFLO(N, L)
         TEMPIN(L) = INTEMP(N, L)
         IF ( QNKWAL ) GO TO 200
         DO 190 J = 1, NQUAL
         QUALIN(L, J) = INQUAL(N, L, J)
 190     CONTINUE
         IF ( QJUNK ) WRITE ( KFILE, WNAME1 )
 200     CONTINUE
*
*  PREPARE OUTFLOW DATA FOR
*  ONE DAY OF SIMULATION
*
         IF ( QWEIR ) WRFLOW = WRFLO(N)
         IF ( QVERI )  GO TO 210
*
*  PREDICTION
*
         TARGET = TARG(N)
         SUMOUT = SUMFLO(N)
         GO TO 240
 210     CONTINUE
*
*  VERIFICATION
*
         OPEN = 0
         SUMOUT = 0.
         IF ( .NOT. QPORT ) GO TO 230
         DO 220 K = 1, NPORTS
         IF ( OUTFLO(N, K) .LE. 0. ) GO TO 220
         OPEN = OPEN + 1
         PHLOW(OPEN) = OUTFLO(N, K)
         SUMOUT = SUMOUT + PHLOW(OPEN)
         PORT(OPEN) = K
 220     CONTINUE
 230     CONTINUE
         IF ( QWEIR ) SUMOUT=SUMOUT + WRFLOW
 240     CONTINUE
*
*  CALL THE SIMULATION SUBROUTINES
*
         IF ( QJUNK ) WRITE ( KFILE, WNAME )
         IF ( .NOT. QNKWAL ) CALL SETTLE
         CALL INFLOW
         IF ( QJUNK ) WRITE ( KFILE, WNAME )
         CALL HEATEX
         IF ( QJUNK ) WRITE ( KFILE, WNAME )
         CALL MIXING
         IF ( QJUNK ) WRITE ( KFILE, WNAME )
         CALL STABLE
         IF ( QJUNK ) WRITE ( KFILE, WNAME )
*
*  COMPLETE SIMULATION CALLS
```

```
*
         IF ( .NOT. QVERI ) CALL DECIDE
         CALL OUTVEL
         IF ( QJUNK ) WRITE ( KFILE, WNAME )
         CALL REFILL
         IF ( QJUNK ) WRITE ( KFILE, WNAME )
         CALL STABLE
         CALL KONVRT
         IF ( QPRINT ) CALL XPRINT
         CALL STABITY
*
*  SAVE RELEASE WATER QUALITY
*  PARAMETER VALUES FOR POSTERITY
*
         AVGT(N) = AVTEMP
         IF ( QNKWAL ) GO TO 260
         DO 250 J = 1, NQUAL
         AVGQ(N, J) = AVQUAL(J)
 250     CONTINUE
 260     CONTINUE
         IF ( QVERI .OR. .NOT. ( QRELE ) ) GO TO 280
         DO 270 K = 1, OPEN
         IJK = PORT(K)
         OUTFLO(N, IJK) = PHLOW(K)
 270     CONTINUE
         POOL = DEPTH
 280     CONTINUE
         IF ( QPRINT ) CALL FINAL
         CALL XCYCLE
         IF ( QVERI ) GO TO 100
*
*  RE-INITIALIZE DATA FOR
*  NEXT YEAR OF SIMULATION
*
         DO 290 K = 1, NFLOOD
         DO 290 N = START, FINISH
         OUTFLO(N, K) = 0.
 290     CONTINUE
         END
```

```
      SUBROUTINE CONVRT
*
*  CONVERT UNITS OF INPUT DATA TO (DEG-C)
*  AND (K-ACRE FT/DAY).  CONVERT UNITS OF
*  FLOW BACK TO (CFS) FOR PRINTOUT
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
      COMMON / BB / TEMP(100), HGT(100), VOLUME(100), TARGET, TMIX, STAB
      COMMON / CH / NM, TITLE
      COMMON / FF / NPORTS, PAREA(8), PHGT(8)
      COMMON / GG / FLOWIN(3),  TEMPIN(3), QUALIN(3, 3)
      COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
      COMMON / JJ / SUMOUT, WIDTH(100)
      COMMON / KK / NWELL(8), FMAX(8), FMIN(8), SELMAX
      COMMON / LL / FGAREA, FGHGT, FGMAX, FGMIN, FGANG, LPORT(8)
      COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
      COMMON / PP / FIRST, LAST, NJ, INDEX(366)
      COMMON / QQ / QVERI, QOUTC, QINTC, QINCFS, QOCFS
      COMMON / RR / INFLO(366, 3), INTEMP(366, 3), INQUAL(366, 3, 3)
      COMMON / SS / OUTFLO(366, 8), TARG(366)
      COMMON / UU / ENFLOW(100), SUMFLO(366), DPRINT(14)
      COMMON / ZZ / DINIT, TINIT(100), QINIT(3, 100)
      COMMON / CD / QPORT, QWEIR, QVINCR, QINITC
      COMMON / GH / WRFLOW, WRFLO(366)
      COMMON / IJ / WANGLE, WTHETA(8)
      CHARACTER TITLE*78, NM*3
      LOGICAL QVERI, QOUTC, QINTC, QINCFS, QOCFS,
     *        QFIRST, QWEIR, QVINCR, QINITC, QWCFS, QPORT
      REAL INFLO, INTEMP, INQUAL
      INTEGER FIRST, OPEN, PORT, DAY
      DATA QFIRST / .TRUE. /
      DATA FACTOR / 1.9835E - 03 /
      DATA SMALL / 1.E - 10 /
      IF ( .NOT. QFIRST ) GO TO 130
      QFIRST = .FALSE.
      IF ( QVERI ) GO TO 130
*
*  IF TARGET TEMPERATURES ARE IN
*  (DEG-F) THEN CONVERT TO (DEG-C)
*
      IF ( QOUTC ) GO TO 110
      DO 100 N = FIRST, LAST
      TARG(N) = 5. / 9. * ( TARG(N) - 32. )
 100  CONTINUE
 110  CONTINUE
*
*  ESTABLISH FLOODGATE PARAMETERS
*
      NFLOOD = NPORTS + 1
      FMAX(NFLOOD) = FGMAX
      FMIN(NFLOOD) = FGMIN
      PHGT(NFLOOD) = FGHGT
      PAREA(NFLOOD) = FGAREA
      WTHETA(NFLOOD) = FGANG
*
*  CONVERT FLOW LIMITS FROM
*  (CFS) TO (K-ACRE FT/DAY)
*
      DO 120 K = 1, NFLOOD
      FMAX(K) = FMAX(K) * FACTOR
      FMIN(K) = FMIN(K) * FACTOR
 120  CONTINUE
```

```
          FGMIN = FGMIN * FACTOR
          FGMAX = FGMAX * FACTOR
          SELMAX = SELMAX * FACTOR
 130      CONTINUE
*
*  IF INFLOW QUANTITIES ARE IN (CFS)
*  THEN CONVERT TO (K-ACRE FT/DAY)
*
          IF ( .NOT. QINCFS ) GO TO 150
          DO 140 N = FIRST, LAST
          DO 140 L = 1, NIP
          INFLO(N, L) = INFLO(N, L) * FACTOR
 140      CONTINUE
 150      CONTINUE
*
*  IF INFLOW TEMPERATURES ARE IN
*  (DEG-F) THEN CONVERT TO (DEG-C)
*
          IF ( QINTC ) GO TO 170
          DO 160 N = FIRST, LAST
          DO 160 L = 1, NIP
          INTEMP(N, L) = 5. / 9. * ( INTEMP(N, L) - 32. )
 160      CONTINUE
 170      CONTINUE
*
*  ZERO ANY NEGATIVE INFLOW
*  TEMPERATURES AND QUANTITIES
*
          DO 180 N = FIRST, LAST
          DO 180 L = 1, NIP
          IF ( INFLO(N, L) .LT. SMALL ) INFLO(N, L) = SMALL
          IF ( INTEMP(N, L) .LT. 0. ) INTEMP(N, L) = 0.
 180      CONTINUE
*
*  IF OUTFLOW QUANTITIES ARE IN (CFS)
*  THEN CONVERT TO (K-ACRE FT/DAY)
*
          IF ( .NOT. QOCFS ) GO TO 220
          IF ( QVFRI ) GO TO 200
*
**  PREDICTION
*
          DO 190 N = FIRST, LAST
          SUMFLO(N) = SUMFLO(N) * FACTOR
 190      CONTINUE
          GO TO 220
*
**  VERIFICATION
*
 200      CONTINUE
          DO 210 N = FIRST, LAST
          DO 210 K = 1, NPORTS
          OUTFLO(N, K) = OUTFLO(N, K) * FACTOR
 210      CONTINUE
 220      CONTINUE
*
*  IF INPUT VOLUMES ARE CUMULATIVE
*  AT THE TOP OF EACH LAYER, THEN
*  COMPUTE INCREMENTAL VOLUMES
*
          IF ( QVINCR ) GO TO 240
          MAX = MAXLAY - 1
```

```
      DO 230 I = 1, MAX
      K = MAXLAY - I + 1
      VOLUME(K) = VOLUME(K) - VOLUME(K - 1)
 230  CONTINUE
 240  CONTINUE
*
*  IF INITIAL TEMPERATURE PROFILE
*  125 IN (DEG-F) THEN CONVERT TO DEG-C
*
      IF ( QINITC ) GO TO 260
      QINITC = .TRUE.
      DO 250 I = 1, MAXLAY
      TINIT(I) = 5. / 9. * ( TINIT(I) - 32. )
 250  CONTINUE
 260  CONTINUE
*
*  IF WEIR FLOWS ARE IN (CFS)
*  THEN CONVERT TO (K-ACRE FT/DAY)
*
      IF ( .NOT. QWEIR .OR. QWCFS ) GO TO 280
      DO 270 N = FIRST, LAST
      WRFLO(N) = WRFLO(N) * FACTOR
 270  CONTINUE
 280  CONTINUE
      RETURN
      ENTRY KONVRT
*
*  CONVERT FLOWS FROM (K-ACRE FT/DAY)
*  TO (CFS) FOR PRINTOUT
*
      SUMOUT = SUMOUT / FACTOR
      DO 290 L = 1, NIP
      FLOWIN(L) = FLOWIN(L) / FACTOR
 290  CONTINUE
      DO 300 K = 1, OPEN
      PHLOW(K) = PHLOW(K) / FACTOR
 300  CONTINUE
      DO 310 I = 1, LSURF
      WTHDRW(I) = WTHDRW(I) / FACTOR
      ENFLOW(I) = ENFLOW(I) / FACTOR
 310  CONTINUE
      RETURN
      END
      SUBROUTINE DECIDE
*
*  THIS SUBROUTINE DETERMINES THE NUMBER AND LOCATION OF
*  PORTS TO USE TO ATTEMPT TO SATISFY THE TARGET TEMPERATURE
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
      COMMON / BB / TEMP(100), HGT(100), VOLUME(100), TARGET, TMIX, STAB
      COMMON / CH / NM, TITLE
      COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
      COMMON / EE / KFILE, LFILE, JFILE, IFILE
      COMMON / FF / NPORTS, PAREA(8), PHGT(8)
      COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
      COMMON / JJ / SUMOUT, WIDTH(100)
      COMMON / KK / NWELL(8), FMAX(8), FMIN(8), SELMAX
      COMMON / LL / FGAREA, FGHGT, FGMAX, FGMIN, FGANG, LPORT(8)
      COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
      COMMON / PP / FIRST, LAST, NJ, INDEX(366)
      COMMON / XX / QJUNK, ONE, TWO
      CHARACTER TITLE*78, NM*3
      DIMENSION XFLOW(3), XPORT(3)
      INTEGER FIRST, PORT, OPEN, XOPEN, XPORT, DAY
      LOGICAL QCHECK, Q1, Q2, QD, QFIRST, QJUNK, QWELL, QMORE
      DIMENSION QWELL(8), QD(8), TPORT(8)
      DATA EXTRA, SMALL, N / 0., 1.E - 08, 1 /
      DATA C / 3.33 /
      DATA QFIRST / .TRUE. /
      QMORE = .TRUE.
      IF ( .NOT. QFIRST ) GO TO 160
      QFIRST = .FALSE.
      NFLOOD = NPORTS + 1
*
*  ORDER PORTS FROM TOP TO BOTTOM
*
      QCHECK = .TRUE.
      X = PHGT(1)
      DO 100 I = 2, NPORTS
      Y = PHGT(I)
      IF ( Y .GT. X ) QCHECK = .FALSE.
      X = Y
 100  CONTINUE
      IF ( QCHECK ) GO TO 130
*
*  EXCHANGE SORT
*
      N = NPORTS - 1
      DO 120 I = 1, N
      K = I + 1
      DO 110 J = K, NPORTS
      IF ( PHGT(I) .GE. PHGT(J) ) GO TO 110
      X = PHGT(I)
      PHGT(I) = PHGT(J)
      PHGT(J) = X
      X = PAREA(I)
      PAREA(I) = PAREA(J)
      PAREA(J) = X
      X = FMAX(I)
      FMAX(I) = FMAX(J)
      FMAX(J) = X
      X = FMIN(I)
      FMIN(I) = FMIN(J)
      FMIN(J) = X
      L = NWELL(I)
```

```
         NWELL(I) = NWELL(J)
         NWELL(J) = L
 110     CONTINUE
 120     CONTINUE
 130     CONTINUE
         J = NWELL(1)
         DO 140 I = 1, NPORTS
         QWELL(I) = NWELL(I) .EQ. J
 140     CONTINUE
         DO 150 I = 1, NFLOOD
         LPORT(I) = 1. + PHGT(I) / DELZ
 150     CONTINUE
 160     CONTINUE
         FLOW = SUMOUT
*
*  RETURN IF THERE NO FLOW
*
         IF ( FLOW + EXTRA .GT. SMALL ) GO TO 180
         EXTRA = 0.
 170     CONTINUE
         OPEN = 1
         PORT(1) = 1
         PHLOW(1) = 1.E - 20
         M = 10
         GO TO 600
 180     CONTINUE
*
*  DETERMINE TEMPERATURE AT EACH PORT
*
         DO 190 I = 1, NFLOOD
         QD(I) = .FALSE.
         IF ( ODEPTH .LT. PHGT(I) ) GO TO 190
         IJK = LPORT(I)
         TPORT(I) = TEMP(IJK)
         QD(I) = .TRUE.
 190     CONTINUE
 200     CONTINUE
*
*  QD(I) = .FALSE. MEANS PORT(I) IS OUTSIDE THE POOL
*
         DO 210 I = 1, NPORTS
         K = I
         IF ( .NOT. QD(I) ) GO TO 210
         IF ( TARGET .GE. TPORT(I) ) GO TO 220
         IF ( TARGET .LE. TPORT(NFLOOD) ) GO TO 310
         GO TO 330
 210     CONTINUE
         IF ( QD(NFLOOD) ) GO TO 310
         EXTRA = FLOW + EXTRA
         FLOW = 0.
         GO TO 170
 220     CONTINUE
*
*  THE FOLLOWING IS EXECUTED IF THE OBJECTIVE
*  TEMPERATURE IS HIGHER THAN THE HIGHEST
*  POSSIBLE WITHDRAWAL TEMPERATURE.  FLOW IS
*  TAKEN FROM THE HIGHEST PORTS
*
         OPEN = 0
         REST = FLOW + EXTRA
         FSLMT = SELMAX
         Q1 = QWELL(K)
```

```
        DO 240 I = K, NPORTS
        IF ( I .EQ. K ) GO TO 230
        IF ( OPEN .EQ. 2 ) GO TO 240
        Q2 = QWELL(I)
        IF ( ( Q1 .AND. Q2 ) .OR. .NOT. ( Q1 .OR. Q2 ) ) GO TO 240
        IF ( OPEN .EQ. 1 ) FSLMT = SELMAX - PHLOW(1)
  230   CONTINUE
        FLW = AMIN1 ( REST, FMAX(I), FSLMT )
        IF ( FLW .LT. FMIN(I) ) GO TO 240
        OPEN = OPEN + 1
        PORT(OPEN) = I
        PHLOW(OPEN) = FLW
        REST = REST - FLW
        IF ( REST .LT. SMALL ) GO TO 250
  240   CONTINUE
        FLW = AMIN1 ( REST, FGMAX )
        IF ( FLW .LT. FGMIN ) GO TO 250
        OPEN = OPEN + 1
        PHLOW(OPEN) = FLW
        PORT(OPEN) = NFLOOD
        REST = REST - FLW
  250   CONTINUE
        EXTRA = REST
        FLOW = FLOW - EXTRA
        IF ( OPEN .GT. 0 ) GO TO 260
        OPEN = 1
        PHLOW(1) = 1.E - 20
        PORT(1) = 1
        M = 15
        GO TO 600
  260   CONTINUE
        GO TO ( 270, 280, 290 ), OPEN
  270   CONTINUE
        M = 20
        GO TO 600
  280   CONTINUE
        IF ( PORT(2) .EQ. NFLOOD ) GO TO 300
        M = 25
        GO TO 600
  290   CONTINUE
        M = 30
        GO TO 600
  300   CONTINUE
        M = 35
        GO TO 600
  310   CONTINUE
*
*  THE FOLLOWING IS EXECUTED IF THE OBJECTIVE
*  TEMPERATURE IS LESS THAN THE LOWEST POSSIBLE
*  WITHDRAWAL TEMPERATURE.  ALL OF THE FLOW IS
*  TAKEN FROM THE FLOOD GATES.
*
        REST = FLOW + EXTRA
        FLW = AMIN1 ( REST, FGMAX )
        IF ( FLW .LT. FGMIN ) GO TO 550
  320   CONTINUE
        EXTRA = REST - FLW
        PORT(1) = NFLOOD
        PHLOW(1) = FLW
        OPEN = 1
        FLOW = FLW
        M = 40
```

```
         GO TO 600
 330     CONTINUE
*
*  THE FOLLOWING IS EXECUTED IF THE OBJECTIVE
*  TEMPERATURE LIES WITHIN SELECTIVE WITHDRAWAL
*  TEMPERATURE LIMITS
*
         REST = FLOW + EXTRA
         IF ( REST .LE. SELMAX ) GO TO 340
         IF ( REST .LT. FGMIN + SELMAX ) GO TO 480
         GO TO 490
 340     CONTINUE
*
*  FLOW IS WITHIN SELECTIVE WITHDRAWAL CAPACITY
*
         DO 350 I = 1, NPORTS
         K = I
         IF ( .NOT. QD(I) ) GO TO 350
         IF ( TARGET .LE. TPORT(I) .AND.
     *        TARGET .GT. TPORT(I + 1) ) GO TO 360
 350     CONTINUE
         EXTRA = REST
         GO TO 170
 360     CONTINUE
         Q1 = QWELL(K)
         K1 = K + 1
         DO 370 I = K1, NFLOOD
         IF ( I .EQ. NFLOOD ) GO TO 370
         IF ( ( Q1 .AND. QWELL(I) ) .OR.
     *        ( .NOT. Q1 .AND. .NOT. QWELL(I) ) ) GO TO 370
         J = I
         GO TO 380
 370     CONTINUE
         J = NFLOOD
         IF ( TPORT(K) - TARGET .GT. TARGET - TPORT(K+1)
     *         .AND. K .NE. NPORTS ) K = K + 1
 380     CONTINUE
         OPEN = 2
         PORT(1) = K
         PORT(2) = J
         IF ( TARGET - TPORT(J) .GT. TPORT(K) - TPORT(J) ) GO TO 384
         PHLOW(1) = REST * ( TARGET - TPORT(J) ) /
     *                     ( TPORT(K) - TPORT(J) )
         GO TO 385
 384     PHLOW(1) = REST
 385     PHLOW(2) = REST - PHLOW(1)
         IF ( PHLOW(1) .GT. FMAX(K) ) GO TO 430
         IF ( PHLOW(1) .LT. FMIN(K) ) GO TO 440
         IF ( PHLOW(2) .LT. FMIN(J) ) GO TO 460
         IF ( PHLOW(2) .GT. FMAX(J) ) GO TO 470
 390     CONTINUE
         EXTRA = 0.
 400     CONTINUE
         FLW = PHLOW(1) + PHLOW(2)
         FLOW = FLW
         IF ( PORT(2) .EQ. NFLOOD ) GO TO 410
         M = 45
         GO TO 420
 410     CONTINUE
         M = 50
 420     CONTINUE
         GO TO 600
```

```
  430     CONTINUE
          PHLOW(1) = FMAX(K)
          PHLOW(2) = REST - PHLOW(1)
          IF ( PHLOW(2) .GE. FMIN(J) ) GO TO 390
          F2 = 0.
          OPEN = 1
          EXTRA = REST - PHLOW(1)
          FLOW = PHLOW(1)
          M = 55
          GO TO 600
  440     CONTINUE
          F1 = PHLOW(1)
          F2 = PHLOW(2)
          PHLOW(1) = FMIN(K)
          PHLOW(2) = AMIN1 ( FMAX(J), REST - PHLOW(1) )
          IF ( PHLOW(2) .GE. FMIN(J) .OR. PHLOW(2) .LT. 0. ) GO TO 450
          PORT(1) = K
          IF ( F2 .GT. F1 ) PORT(1) = J
          OPEN = 1
          EXTRA = 0.
          PHLOW(1) = REST
          M = 60
          GO TO 600
  450     CONTINUE
          IF ( PHLOW(2) .GE. 0. ) GO TO 390
          EXTRA = PHLOW(2)
          PHLOW(2) = 0.
          OPEN = 1
          GO TO 400
  460     CONTINUE
          PHLOW(2) = 0.
          PHLOW(1) = AMIN1 ( FMAX(K), REST - PHLOW(2) )
          OPEN = 1
          GO TO 390
  470     CONTINUE
          PHLOW(2) = FMAX(J)
          PHLOW(1) = REST - PHLOW(2)
          GO TO 390
  480     CONTINUE
*
*   FLOW IS GREATER THAN SELECTIVE WITHDRAWAL
*   CAPACITY BY LESS THAN FLOOD GATE MINIMUM
*   CAPACITY.  ONLY SELECTIVE WITHDRAWAL
*   MAXIMUM IS WITHDRAWN.
*
          REST = SELMAX
          GO TO 340
  490     CONTINUE
*
*   FLOW IS LARGE ENOUGH TO REQUIRE THAT
*   SOME IS TAKEN FROM FLOOD GATE.
*
          FLW = REST - SELMAX
          TX = ( TARGET * REST - TPORT(NFLOOD) * FLW ) / ( REST - FLW )
          QCHECK = .TRUE.
          DO 500 I = 1, NPORTS
          IF ( .NOT. QD(I) ) GO TO 500
          K = I
          IF ( QCHECK .AND. TX .GE. TPORT(I) ) GO TO 520
          QCHECK = .FALSE.
          IF ( .NOT. ( TX .LT. TPORT(I) .AND.
     *         TX .GE. TPORT(I + 1) ) ) GO TO 500
```

```
         GO TO 510
 500     CONTINUE
         OPEN = 1
         PORT(1) = NFLOOD
         PHLOW(1) = REST
         EXTRA = 0.
         FLOW = PHLOW(1)
         M = 65
         GO TO 600
 510     CONTINUE
         OPEN = 2
         PORT(1) = K
         PORT(2) = NFLOOD
         PHLOW(2) = REST * ( TARGET - TPORT(K) ) /
     *              ( TPORT(NFLOOD) - TPORT(K) )
         PHLOW(1) = AMIN1 ( FMAX(K), REST - PHLOW(2) )
         IF ( PHLOW(1) .LT. FMIN(K) ) PHLOW(1) = 0.
         PHLOW(2) = REST - PHLOW(1)
         EXTRA = 0.
         M = 70
         GO TO 600
 520     CONTINUE
*
*  THE OBJECTIVE TEMPERATURE IS HIGHER THAN
*  THE HIGHEST EMPERATURE WHICH CAN BE OBTAINED
*  FOR THE GIVEN FLOW.  RELEASE IS FROM THE TOP
*  PORTS AND THE FLOOD GATE.
*
         PHLOW(1) = FMAX(K)
         PHLOW(2) = 0.
         OPEN = 1
         PORT(1) = K
         REST = REST - PHLOW(1)
         QCHECK = QWELL(K)
         DO 530 I = K, NPORTS
         IF ( ( QCHECK .AND. QWELL(I) )  .OR.
     *        .NOT. ( QCHECK .OR. QWELL(I) ) ) GO TO 530
         PHLOW(2) = AMIN1 ( SELMAX - PHLOW(1), FMAX(I) )
         PORT(2) = I
         OPEN = 2
         IF ( PHLOW(2) .GT. FMIN(I) ) GO TO 540
         OPEN = 1
         PHLOW(2) = 0.
         GO TO 540
 530     CONTINUE
 540     CONTINUE
         REST = REST - PHLOW(2)
         OPEN = OPEN + 1
         PHLOW(OPEN) = AMIN1 ( FGMAX, REST )
         PORT(OPEN) = NFLOOD
         EXTRA = REST - PHLOW(OPEN)
         FLOW = FLOW - EXTRA
         N = OPEN - 1
         M = 80
         GO TO 600
 550     CONTINUE
*
*  THE FOLLOWING IS EXECUTED IF THE OBJECTIVE TEMPERATURE
*  IS LESS THAN THE LOWEST POSSIBLE WITHDRAWAL TEMPERATURE
*  BUT THE REQUIRED FLOW IS LESS THAN THE FLOOD GATE
*  MINIMUM CAPACITY THE FLOW IS TAKEN FROM THE LOWEST
*  SELECTIVE WITHDRAWAL PORTS IF POSSIBLE
```

```
*
         I = NPORTS
         PHLO = AMIN1 ( FMAX(I), FLW )
         IF ( PHLO .GE. FMIN(I) ) GO TO 560
         FLW = 0
         GO TO 320
 560     CONTINUE
         PORT(1) = I
         PHLOW(1) = PHLO
         PHLOW(2) = 0.
         OPEN = 1
         PHLO = FLW - PHLO
         IF ( PHLO .LE. 0. ) GO TO 590
         Q1 = QWELL(I)
         DO 570 J = 1, NPORTS
         K = NPORTS - J + 1
         Q2 = QWELL(K)
         IF ( ( Q1 .AND. Q2 ) .OR. .NOT. ( Q1 .OR. Q2 ) ) GO TO 570
         IF ( .NOT. QD(K) ) GO TO 590
         GO TO 580
 570     CONTINUE
         GO TO 590
 580     CONTINUE
         PHLO = AMIN1 ( PHLO, FMAX(K) )
         IF ( PHLO .LT. FMIN(K) ) GO TO 590
         PORT(2) = K
         PHLOW(2) = PHLO
         OPEN = 2
 590     CONTINUE
         FLOW = PHLOW(1) + PHLOW(2)
         EXTRA = REST - FLOW
         M = 85
 600     CONTINUE
         INDEX(DAY) = M
         IF ( .NOT. QMORE ) GO TO 650
         QMORE = .FALSE.
         IF ( M .LT. 45 .OR. M .GT. 60 ) GO TO 650
*
*  COMPUTE RELEASE TEMPERATURE OF FLOW
*  THROUGH SELECTED PORT CONFIGURATION
*  AND AGAIN DETERMINE PORTS AND FLOWS
*
         XOPEN = OPEN
         DO 610 K = 1, XOPEN
         XFLOW(K) = PHLOW(K)
         XPORT(K) = PORT(K)
 610     CONTINUE
         DO 640 K = 1, XOPEN
         J = XPORT(K)
         OPEN = 1
         PHLOW(1) = XFLOW(K)
         PORT(1) = XPORT(K)
*
*  COMPUTE VELOCITY FOR ONE OPEN PORT
*
         CALL OUTVEL
*
*  COMPUTE RELEASE TEMPERATURE
*  FOR ONE OPEN PORT
*
         SUM = 0.
         DO 620 I = 1, LSURF
```

```
      WTHDRW(I) = VEL(I) * WIDTH(I)
      SUM = SUM + WTHDRW(I)
620   CONTINUE
      SCALE = PHLOW(1) / SUM
      SUMTF = 0.
      DO 630 I = 1, LSURF
      WTHDRW(I) = WTHDRW(I) * SCALE
      SUMTF = SUMTF + WTHDRW(I) * TEMP(I)
630   CONTINUE
      TNEW = SUMTF / PHLOW(1)
      TPORT(J) = TNEW
640   CONTINUE
      GO TO 200
650   CONTINUE
      RETURN
      END
```

```
         FUNCTION DENINT ( X )
*
*  INTERPOLATE TO DETERMINE
*  DENSITY AT ANY LOCATION
*
         COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
     *          / CC / DEN(100), NUSURF
     *          / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
     *          / EF / AREA, HGTPRT, LAYPRT, DENPRT
     *          / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
*
         INTEGER SIGN
*
         LOGICAL QDEN, QTLIM, QBLIM, QSINK1, QSINK2, QSHIFT, QWEIR *
         DATA SMALL / 1.E - 05 /
*
         LAYER = 1. + X / DELZ
*
         IF ( X .GE. DEPTH .OR. X .LT. 0.0 ) GO TO 120
*
*.... IF THE LAYER IS OUTSIDE THE POOL, THE DENSITY IS
*     EXTRAPOLATED BASED ON A LINEAR DENSITY GRADIENT EXTENDED
*     FROM THE PORT CENTERLINE TO THE DESIRED BOUNDARY LAYER ....
*
*.... FIND THE DENSITY INSIDE THE POOL ....
*
         ELMID = DELZ * ( FLOAT ( LAYER ) - 0.5 )
         DIFF = ABS ( ELMID - X )
         IF ( DIFF .LT. SMALL ) THEN
            DENINT = DEN ( LAYER )
            RETURN
         ENDIF
*
         IF ( LAYER .EQ. LSURF .AND. X .GE. ELMID ) THEN
            SLOPE = ( DEN ( LSURF - 1 ) - DEN ( LSURF ) )
     &                / DELZ
            DENINT = DEN ( LAYER ) - DIFF * SLOPE
            RETURN
         ELSEIF ( LAYER .EQ. 1 .AND. X .LE. ELMID ) THEN
            SLOPE = ( DEN ( 1 ) - DEN ( 2 ) ) / DELZ
            DENINT = DEN ( LAYER ) + DIFF * SLOPE
            RETURN
         ENDIF
*
         SIGN = ( ELMID - X ) / ABS ( ELMID - X )
         IJK = - ( SIGN - 1 ) / 2
         IJ = LAYER + IJK
         JK = IJ - 1
         SLOPE = ( DEN ( IJ ) - DEN ( JK ) ) / DELZ
         ELTOP = DELZ * ( FLOAT ( IJ ) - 0.5 )
         DENINT = DEN ( IJ ) - ( ELTOP - X ) * SLOPE
         RETURN
 120     CONTINUE
*
*.... FIND THE DENSITY OUTSIDE THE POOL ....
*
         IF ( HGTPRT .GE. DEPTH - 0.5 * DELZ ) THEN
            DGRDT = ( DEN ( LSURF ) - DEN ( LSURF - 1 ) ) / DELZ
         ELSE
            DGRDT = ( DEN ( LSURF ) - DENPRT ) / ( DEPTH - HGTPRT )
         ENDIF
*
```

```
IF ( HGTPRT .LE. 0.5 * DELZ ) THEN
   DGRDB = ( DEN ( 1 ) - DEN ( 2 ) ) / DELZ
ELSE
   DGRDB = ( DEN ( 1 ) - DENPRT ) / HGTPRT
ENDIF
IF ( LAYER .GE. LSURF ) DGRD = DGRDT
IF ( LAYER .LE. 1 ) DGRD = DGRDB
DENINT = DGRD * ABS ( HGTPRT - X ) + DENPRT
RETURN
END
```

```
        SUBROUTINE DETAIL
*
*  PRINT INTERESTING DATA
*  FOR SPECIAL PRINT DAYS
*
        COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
        COMMON / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
        COMMON / CH / NM, TITLE
        COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
        COMMON / EE / KFILE, LFILE, JFILE, IFILE
        COMMON / FF / NPORTS, PAREA(8), PHGT(8)
        COMMON / GG / FLOWIN(3), TEMPIN(3), QUALIN(3, 3)
        COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
        COMMON / II / EK, ET, SHORT, BETA, LAMDA, MIXCOEF, GAMMA,
     *                 DECAY
        COMMON / JJ / SUMOUT, WIDTH(100)
        COMMON / LL / FGAREA, FGHGT, FGMAX, FGMIN, FGANG, LPORT(8)
        COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
        COMMON / PP / FIRST, LAST, NJ, INDEX(366)
        COMMON / OO / AVTEMP, AVQUAL(3), AVGT(366), AVGQ(366, 3)
        COMMON / QQ / QVERI, QOUTC, QINTC, QINCFS, QOCFS, QWCFS
        COMMON / UU / ENFLOW(100), SUMFLO(366), DPRINT(14)
        COMMON / WW / QNKWAL, YEAR(2)
        CHARACTER TITLE*78
        LOGICAL QVERI, QNKWAL
        DIMENSION NUMBER(12)
        INTEGER PORT, OPEN, DAY, FIRST, YEAR
        CHARACTER*1 BLANK, XXXX
        CHARACTER*3 NM, MONTH(12)*3, LEVEL(100)*1
        DATA BLANK, XXXX / ' ', 'X' /
        DATA MONTH / 'JAN', 'FEB', 'MAR', 'APR',
     *               'MAY', 'JUN', 'JUL', 'AUG',
     *               'SEP', 'OCT', 'NOV', 'DEC' /
        DATA NUMBER / 31, 28, 31, 30, 31, 30,
     *                31, 31, 30, 31, 30, 31 /
*
*  DETERMINE AND PRINT DATE INFORMATION
*
        M = 0
 100    CONTINUE
        DO 110 K = 1, 12
        M = M + NUMBER(K)
        IF ( DAY .GT. M ) GO TO 110
        NJ = NUMBER(K) + DAY - M
        NM = MONTH(K)
        GO TO 120
 110    CONTINUE
        GO TO 100
 120    CONTINUE
        WRITE ( LFILE, 500 ) TITLE
        IF ( YEAR(1) .GT. 0 ) WRITE ( LFILE, 510 ) YEAR
        WRITE ( LFILE, 520 ) DAY, NJ, NM
*
*  PRINT INPUT DATA AND
*  COMPUTED VALUES
*
        DO 140 L = 1, NIP
        WRITE ( LFILE, 530 ) FLOWIN(L)
        WRITE ( LFILE, 540 ) TEMPIN(L)
        IF ( QNKWAL ) GO TO 140
        DO 130 J = 1, NQUAL
        WRITE ( LFILE, 550 ) J, QUALIN(L, J)
 130     CONTINUE
 140     CONTINUE
         WRITE ( LFILE, 560 ) ET
         WRITE ( LFILE, 570 ) EK
         WRITE ( LFILE, 580 ) SHORT
         WRITE ( LFILE, 590 ) DEPTH
         IF ( .NOT. QVERI ) WRITE ( LFILE, 600 ) TARGET
         WRITE ( LFILE, 610 ) AVTEMP
         IF ( QNKWAL ) GO TO 160
         DO 150 J = 1, NQUAL
         WRITE ( LFILE, 620 ) J, AVQUAL(J)
 150     CONTINUE
 160     CONTINUE
         WRITE ( LFILE, 630 ) SUMOUT
         DO 170 K = 1, OPEN
         WRITE ( LFILE, 640 ) PORT(K), PHLOW(K)
 170     CONTINUE
*
*  DETERMINE LEVELS OF OPEN PORTS
*
         DO 180 I = 1, LSURF
         LEVEL(I) = BLANK
 180     CONTINUE
         DO 190 K = 1, OPEN
         IJK = PORT(K)
         JKL = 1. + PHGT(IJK) / DELZ
         LEVEL(JKL) = XXXX
 190     CONTINUE
*
*  PRINT HEADING
*
         WRITE ( LFILE, 500 ) TITLE
         IF ( YEAR(1) .GT. 0 ) WRITE ( LFILE, 510 ) YEAR
         WRITE ( LFILE, 520 ) DAY, NJ, NM
         WRITE ( LFILE, 650 )
         IF ( QNKWAL ) GO TO 210
         DO 200 J = 1, NQUAL
         IF ( J .EQ. 1 ) WRITE ( LFILE, 660 ) J
         IF ( J .EQ. 2 ) WRITE ( LFILE, 670 ) J
         IF ( J .EQ. 3 ) WRITE ( LFILE, 680 ) J
 200     CONTINUE
 210     CONTINUE
         WRITE ( LFILE, 690 )
*
*  PRINT INFORMATION TABLES
*
         DEEP = - DELZ
         LSURFP1 = LSURF + 1
         DO 212 M = LSURFP1, MAXLAY
         ENFLOW(LSURF)=ENFLOW(LSURF) + ENFLOW(M)
 212     CONTINUE
         DO 220 M = 1, LSURF
         I = LSURF - M + 1
         DEEP = DEEP + DELZ
         ELEV = ELEV - DELZ
         IF ( I .EQ. 1 ) ELEV = BOTTOM
         IF ( I .EQ. LSURF ) ELEV = BOTTOM + DEPTH
         IF ( I .EQ. LSURF - 1 ) ELEV = BOTTOM + DEPTH -
     *                 DELZ * ( HGT(LSURF) + 0.5 )
         WRITE ( LFILE, 690 ) LEVEL(I), I, ELEV, DEEP, ENFLOW(I),
     *                        WTHDRW(I), VEL(I), TEMP(I)
         IF ( QNKWAL ) GO TO 220
```

```
           WRITE ( LFILE, 700 ) ( QUAL(J, I), J = 1, NQUAL )
  220      CONTINUE
           RETURN
  500      FORMAT ( '1' // 20X, A )
  510      FORMAT ( 20X, I4, 'H - HYDROLOGY' /
     *              20X, I4, 'H - METEOROLOGY' )
  520      FORMAT ( /// 20X, 'DAY - ', I3,
     *                   5X, I2, 1X, A3 // )
  530      FORMAT ( // 20X, 'INFLOW QUANTITY',
     *                 15X, F10.2, 5X, 'CFS' )
  540      FORMAT (  / 20X, 'INFLOW TEMPERATURE',
     *                 12X, F10.2, 5X, 'DEG-C' )
  550      FORMAT (  / 20X, 'INFLOW QUALITY - ', I1,
     *                 12X, F10.2, 5X, 'MG/L' )
  560      FORMAT ( // 20X, 'EQUILIBRIUM TEMPERATURE',
     *                  7X, F10.2, 5X, 'DEG-F' )
  570      FORMAT (  / 20X, 'HEAT EXCHANGE COEFFICIENT',
     *                  5X, F10.2, 5X, 'BTU/DEG-F' )
  580      FORMAT (  / 20X, 'SHORT WAVE RADIATION',
     *                 10X, F10.2, 5X, 'BTU' )
  590      FORMAT (  / 20X, 'POOL ELEVATION',
     *                 16X, F10.2, 5X, 'FEET' )
  600      FORMAT ( // 20X, 'TARGET TEMPERATURE',
     *                 12X, F10.2, 5X, 'DEG-C' )
  610      FORMAT (  / 20X, 'RELEASE TEMPERATURE',
     *                 11X, F10.2, 5X, 'DEG-C' )
  620      FORMAT (  / 20X, 'RELEASE QUALITY - ', I1,
     *                 11X, F10.2, 5X, 'MG/L' )
  630      FORMAT (  / 20X, 'OUTFLOW QUANTITY',
     *                 14X, F10.2, 5X, 'CFS' )
  640      FORMAT (  / 20X, 'OUTFLOW PORT - ', I1,
     *                 14X, F10.2, 5X, 'CFS' )
  650      FORMAT ( // 15X, 'LAYER', 3X, 'ELEVATION',
     *                  3X, 'DEPTH', 3X, 'INFLOW',
     *                  3X, 'WITHDRAWAL', 3X, 'VELOCITY',
     *                  3X, 'TEMPERATURE' )
  660      FORMAT ( '+',  90X, 'QUALITY-', I1 )
  670      FORMAT ( '+', 102X, 'QUALITY-', I1 )
  680      FORMAT ( '+', 114X, 'QUALITY-', I1 )
  690      FORMAT (     9X, A1, 7X, I2, 1X, F10.1,
     *                  1X, F9.1, 1X, F8.0, 1X, F10.2,
     *                  1X, F10.2, 3X, F10.2 )
  700      FORMAT ( '+', 84X, 3 ( 3X, F10.1 ) )
           END
```

```
        SUBROUTINE ERROR ( IJK )
*
*  OUTPUTS ERROR CODE TO ASSIST IN DETERMINING THE LOCATION WITHIN
*  THE CODE THAT THE ERROR OCCURRED
*
        COMMON / EE / KFILE, LFILE, JFILE, IFILE
        WRITE ( LFILE, 500 ) IJK
        STOP
 500    FORMAT ( /// 3 ( 5X, '***' / ), 5X,
     *             '***', 5X, 'STOP CODE - ',
     *             I4 / 3( 5X, '***' / ) )
        END
```

```
      SUBROUTINE FINAL
*
*  OUTPUTS THE FINAL SUMMARY INFORMATION
*  INCLUDING RELEASE TEMPERATURE AND QUALITY, PORT
*  SELECTION INDICES (PREDICTION) AND RELEASE TEMPERATURE
*  STATISTICS (PREDICTION)
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
      COMMON / CH / NM, TITLE
      COMMON / EE / KFILE, LFILE, JFILE, IFILE
      COMMON / OO / AVTEMP, AVQUAL(3), AVGT(366), AVGQ(366, 3)
      COMMON / PP / FIRST, LAST, NJ, INDEX(366)
      COMMON / QQ / QVERI, QOUTC, QINTC, QINCFS, QOCFS
      COMMON / SS / OUTFLO(366, 8), TARG(366)
      COMMON / VV / START, FINISH, QRELE, QPROF
      COMMON / WW / QNKWAL, YEAR(2)
      DIMENSION X(10), L(10), B(10), DIFF(366)
      CHARACTER TITLE*78, NM*3, DUMMY*1
      INTEGER FIRST, B, START, FINISH, YEAR
      REAL MXDIFF, MXGRAD
      LOGICAL QVERI, QNKWAL
      INCR = 1 + ( FINISH - START + 1 ) / 8
      IF ( START .EQ. 1 ) INCR = 50
      IF( FINISH .LT. INCR ) INCR = FINISH
*
*  OUTPUT RELEASE TEMPERATURES
*
      WRITE ( LFILE, 560 ) TITLE
      IF ( YEAR(1) .GT. 0 ) WRITE ( LFILE, 570 ) YEAR
      WRITE ( LFILE, 500 )
      DO 120 M = 1, INCR
      K = 0
      I = M + START - 1
      DO 110 N = I, FINISH, INCR
      K = K + 1
      L(K) = N
      X(K) = AVGT(N)
  110 CONTINUE
      WRITE ( LFILE, 530 ) ( L(J), X(J), J = 1, K )
  120 CONTINUE
*
*  OUTPUT RELEASE QUALITIES
*
      IF ( QNKWAL ) GO TO 160
      DO 150 J = 1, NQUAL
      WRITE ( LFILE, 560 ) TITLE
      IF ( YEAR(1) .GT. 0 ) WRITE ( LFILE, 570 ) YEAR
      IF ( NQUAL .GT. 1 ) WRITE ( LFILE, 510 ) J
      IF ( NQUAL .EQ. 1 ) WRITE ( LFILE, 520 )
      DO 140 M = 1, INCR
      K = 0
      I = M + START - 1
      DO 130 N = I, FINISH, INCR
      K = K + 1
      L(K) = N
      X(K) = AVGQ(N, J)
  130 CONTINUE
      WRITE ( LFILE, 530 ) ( L(N), X(N), N = 1, K )
  140 CONTINUE
  150 CONTINUE
  160 CONTINUE
      IF ( QVERI ) GO TO 220
```

```
*
*  PRINT PORT SELECTION INDICES
*
        WRITE ( LFILE, 560 ) TITLE
        IF ( YEAR(1) .GT. 0 ) WRITE ( LFILE, 570 ) YEAR
        WRITE ( LFILE, 540 )
        DO 180 M = 1, INCR
        K = 0
        I = M + START - 1
        DO 170 N = I, FINISH, INCR
        K = K + 1
        L(K) = N
        B(K) = INDEX(N)
 170    CONTINUE
        WRITE ( LFILE, 550 ) ( L(J), B(J), J = 1, K )
 180    CONTINUE
*
*  OUTPUT DIFFERENCES OF RELEASE
*  AND TARGET TEMPERATURES
*
        WRITE ( LFILE, 560 ) TITLE
        IF ( YEAR(1) .GT. 0 ) WRITE ( LFILE, 570 ) YEAR
        WRITE ( LFILE, 580 )
        DO 200 M = 1, INCR
        K = 0
        I = M + START - 1
        DO 190 N = I, FINISH, INCR
        K = K + 1
        L(K) = N
        X(K) = TARG(N) - AVGT(N)
 190    CONTINUE
        WRITE ( LFILE, 530 ) ( L(J), X(J), J = 1, K )
 200    CONTINUE
*
* OUTPUT RELEASE TEMPERATURE STATISTICS
*
        SMDIFF = 0.
        SADIFF = 0.
        MXDIFF = 0.
        MXGRAD = 0.
        NUMBER = FINISH - START + 1
        DO 210 N = START, FINISH
        DIFF(N) = AVGT(N) - TARG(N)
        SMDIFF = SMDIFF + DIFF(N)
        SADIFF = SADIFF + ABS( DIFF(N) )
        SSDIFF = SSDIFF + DIFF(N) ** 2
        MXDIFF = AMAX1 ( MXDIFF, ABS ( DIFF(N) ) )
        IF ( N .EQ. START ) GO TO 210
        MXGRAD = AMAX1 ( MXGRAD, ABS ( DIFF(N) - DIFF(N - 1) ) )
 210    CONTINUE
        AVDIFF = SMDIFF / FLOAT ( NUMBER )
        AADIFF = SADIFF / FLOAT ( NUMBER )
        WRITE ( LFILE, 560 )
        WRITE ( LFILE, 590 ) SMDIFF
        WRITE ( LFILE, 600 ) SADIFF
        WRITE ( LFILE, 610 ) SSDIFF
        WRITE ( LFILE, 620 ) MXDIFF
        WRITE ( LFILE, 630 ) MXGRAD
        WRITE ( LFILE, 640 ) AVDIFF
        WRITE ( LFILE, 650 ) AADIFF
 220    CONTINUE
        RETURN
```

```
500     FORMAT ( // 10X, '***  RELEASE TEMPERATURE',
    *                    '  ( DEG-C )  ***' /// )
510     FORMAT ( // 10X, '***  RELEASE QUALITY - ', I1,
    *                    '  ( MG/L )  ***' /// )
520     FORMAT ( // 10X, '***  RELEASE QUALITY ',
    *                  ' ( MG/L )  ***' /// )
530     FORMAT ( 8 ( 6X, I3, 1X, F6.1 ) )
540     FORMAT ( // 10X, '***  PORT SELECTION INDICES  ***' /// )
550     FORMAT ( 8 ( 6X, I3, 1X, I6 ) )
560     FORMAT ( '1' / 10X, A )
570     FORMAT ( 10X, I4, ' - HYDROLOGY' /
    *            10X, I4, ' - METEOROLOGY' )
580     FORMAT ( // 10X, '***  TEMPERATURE DIFFERENCE',
    *                  '  ( RELEASE - TARGET )  ***' /// )
590     FORMAT ( /// 7X, 'SUM OF DIFFERENCES', 22X, 1PG12.2 )
600     FORMAT ( /   7X, 'SUM OF ABSOLUTE DIFFERENCES', 13X, 1PG12.2 )
610     FORMAT ( /   7X, 'SUM OF SQUARES DIFFERENCES', 14X, 1PG12.2 )
620     FORMAT ( /   7X, 'MAXIMUM DIFFERENCE', 22X, 1PG12.2 )
630     FORMAT ( /   7X, 'MAXIMUM 1-DAY TEMPERATURE CHANGE',
    *                8X, 1PG12.2 )
640     FORMAT ( /   7X, 'AVERAGE DIFFERENCE', 22X, 1PG12.2 )
650     FORMAT ( /   7X, 'AVERAGE ABSOLUTE DIFFERENCE', 13X, 1PG12.2 )
660     FORMAT ( I10, 2F10.2)
        END
        SUBROUTINE HEATEX
*
*  COMPUTE HEAT EXCHANGE AT THE AIR-WATER INTERFACE
*  AND DISTRIBUTION SHORT WAVE RADIATION WITHIN THE POOL
*
        COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
        COMMON / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
        COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
        COMMON / EE / KFILE, LFILE, JFILE, IFILE
        COMMON / II / EK, ET, SHORT, BETA, LAMBDA, MIXCOEF, GAMMA,
     *               DECAY
        COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
        COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
        COMMON / XX / QJUNK, ONE, TWO
        DIMENSION HEAT(100)
        REAL LAMBDA
        INTEGER DAY
        LOGICAL*4 QJUNK, QFIRST, QFEET, QMETR
        DATA QFEET, QMETR / .TRUE., .FALSE. /
        DATA QFIRST / .TRUE. /
        DATA RHO / 62.4 /
        DATA SMALL / 1.0E-10 /
*
*  ESTABLISH DEPTH IN WHICH SURFACE
*  HEAT EXCHANGE TAKES PLACE
*
        IF ( .NOT. QFIRST ) GO TO 100
        QFIRST = .FALSE.
        IF ( QFEET ) HDEPTH = 2.
        IF ( QMETR ) HDEPTH = .6096
 100    CONTINUE
*
*  COMPARE SURFACE HEAT EXCHANGE
*  DEPTH TO THICKNESS OF TOP LAYER
*
        THETA = 9. / 5. * TEMP(LSURF) + 32.
        HTOTAL = EK * ( ET - THETA )
        HDOWN = ( 1. - BETA ) * SHORT
        TOP = HGT(LSURF) * DELZ
        IF ( TOP .GE. HDEPTH ) GO TO 110
        IF ( TOP .LT. HDEPTH ) GO TO 120
 110    CONTINUE
*
*  COMPUTE HEAT TRANSFER INTO
*  A LARGE SURFACE LAYER
*
        HSURF = HTOTAL - HDOWN
        EXTRA = HDOWN * ( EXP ( - LAMBDA * HDEPTH ) -
     *          EXP ( - LAMBDA * TOP ) )
        HEAT(LSURF) = HSURF + EXTRA
        HDOWN = HDOWN - EXTRA
        LSM = LSURF - 1
        GO TO 150
 120    CONTINUE
*
*  COMPUTE SURFACE HEAT TRANSFER
*  WITH A SMALL SURFACE LAYER
*
        IF ( TOP + DELZ .GT. HDEPTH ) GO TO 130
        IF ( TOP + 2. * DELZ .GT. HDEPTH ) GO TO 140
        CALL ERROR ( 2000 )
 130    CONTINUE
```

```
*
*  SURFACE HEAT EXTENDS
*  INTO SECOND LAYER
*
      HEAT(LSURF) = ( HTOTAL - HDOWN ) * TOP / HDEPTH
      HSM1 = ( HTOTAL - HDOWN ) * ( 1. - TOP / HDEPTH )
      BNEXT = TOP + DELZ
      EXTRA = HDOWN * ( EXP ( - LAMBDA * HDEPTH ) -
     *        EXP ( - LAMBDA * BNEXT ) )
      HEAT(LSURF - 1) = HSM1 + EXTRA
      HDOWN = HDOWN - EXTRA
      LSM = LSURF - 2
      GO TO 150
 140  CONTINUE
*
*  SURFACE HEAT DEPTH EXTENDS
*  INTO THIRD LAYER
*
      HEAT(LSURF) = ( HTOTAL - HDOWN ) * TOP / HDEPTH
      HEAT(LSURF - 1) = ( HTOTAL - HDOWN ) * DELZ / HDEPTH
      HSM2 = ( HTOTAL - HDOWN ) * ( 1. - ( TOP + DELZ ) / HDEPTH )
      BNEXT = TOP + 2. * DELZ
      EXTRA = HDOWN * ( EXP ( - LAMBDA * HDEPTH ) -
     *        EXP ( - LAMBDA * BNEXT ) )
      HDOWN = HDOWN - EXTRA
      HEAT(LSURF - 2) = HSM2 + EXTRA
      LSM = LSURF - 3
 150  CONTINUE
*
*  COMPUTE HEAT TRANSFER
*  INTO ALL OTHER LAYERS
*
      SUM = 0.
      DO 160 I = 1, LSM
      ZMID = DEPTH - ( FLOAT ( I ) * DELZ - DELZ / 2. )
      HEAT(I) = ( 1. - BETA ) * SHORT * EXP ( - LAMBDA * ZMID )
     *          * LAMBDA * DELZ
      SUM = SUM + HEAT(I)
 160  CONTINUE
*
*  PROPORTION HEAT OF EACH LAYER
*  TO TOTAL HEAT ENTERING POOL
*
       IF ( SUM .LT. SMALL ) THEN
       SCALE = 1.0
       ELSE
       SCALE = HDOWN / SUM
       END IF
      DO 170 I = 1, LSM
      HEAT(I) = HEAT(I) * SCALE
 170  CONTINUE
*
*  COMPUTE CHANGE OF TEMPERATURE IN EACH
*  LAYER DUE TO SURFACE HEAT EXCHANGE
*
      DO 180 I = 1, LSURF
      CHANGE = 5. / 9. * HEAT(I) / ( RHO * DELZ * HGT(I) )
      TEMP(I) = TEMP(I) + CHANGE
      IF ( QJUNK )  WRITE( KFILE,530)  I, HEAT(I), TEMP(I), CHANGE
 180  CONTINUE
      IF ( QJUNK ) WRITE ( KFILE, 500 )
      IF ( QJJNK ) WRITE ( KFILE, 510 ) DAY
```

```
        IF ( QJUNK ) WRITE ( KFILE, 520 )
    *                ( I, TEMP(I), I = 1, LSURF )
        RETURN
500     FORMAT ( '1' )
510     FORMAT ( /// 5X, 'DAY - ', I3, 5X, 'AFTER HEAT' )
530     FORMAT ( I11, F12.2, F11.2,  F9.3 )
520     FORMAT ( /// ( 8 ( 5X, I2, 2X, F6.1 ) ) )
        END
```

```
         SUBROUTINE HMREAD
*
*  READ ALL HYDROLOGIC
*  AND METEOROLOGICAL DATA
*
         COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
         COMMON / CH / NM, TITLE
         COMMON / EE / KFILE, LFILE, JFILE, IFILE
         COMMON / FF / NPORTS, PAREA(8), PHGT(8)
         COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
         COMMON / PP / FIRST, LAST, NJ, INDEX(366)
         COMMON / QQ / QVERI, QOUTC, QINTC, QINCFS, QOCFS, QWCFS
         COMMON / RR / INFLO(366, 3), INTEMP(366, 3), INQUAL(366, 3, 3)
         COMMON / SS / OUTFLO(366, 8), TARG(366)
         COMMON / TT / EQTEMP(366), EXCOEF(366), SOLAR(366)
         COMMON / UU / ENFLOW(100), SUMFLO(366), DPRINT(14)
         COMMON / WW / QNKWAL, YEAR(2)
         COMMON / XX / QJUNK, ONE, TWO
         COMMON / CD / QPORT, QWEIR, QVINCR, QINITC
         COMMON / GH / WRFLOW, WRFLO(366)
         COMMON / BL / WIND(366)
         DIMENSION NUMBER(13), VALUE(12)
         INTEGER YEAR , FIRST, DFIRST, DLAST
         CHARACTER TITLE*78, NM*3
         CHARACTER*4 XSTOP, XINFL, XFAHR, XCELS, BLANK, XEQUI,
     *           XEXCH, XSHOR, XKACF, XCFS, XOUTF, XDEPT,
     *           XTEMP, XEND,  XMONT, XSINC, XQUAL, XWEIR,
     *           CHECK, UNITS, TYPE, XWIND
         LOGICAL *4 QINCFS, QINTC, QVERI, QOCFS, QWCFS,
     *           QPORT, QWEIR, QNKWAL, QFIRST, QJUNK
         REAL INFLO, INTEMP, INQUAL
         DATA NUMBER / 0, 31, 59, 90, 120, 151, 181,
     *                 212, 243, 272, 304, 334, 365 /
         DATA XSTOP, XINFL, XFAHR / 'STOP', 'INFL', 'FAHR' /
         DATA BLANK, XQUAL, XWEIR / '    ', 'QUAL', 'WEIR' /
         DATA XEQUI, XEXCH, XSHOR / 'EQUI', 'EXCH', 'SHOR' /
         DATA XMONT, XSINC, XCELS / 'MONT', 'SINC', 'CELS' /
         DATA XKACF, XCFS , XOUTF / 'KACF', 'CFS ', 'OUTF' /
         DATA XDEPT, XTEMP, XEND  / 'DEPT', 'TEMP', 'END ' /
         DATA XWIND / 'WIND' /
         DATA QFIRST / .TRUE. /
         IF ( .NOT. QFIRST ) GO TO 110
         QFIRST = .FALSE.
*
*  DETERMINE RANGE OF MONTHS TO
*  INCLUDE DATA INPUT INTERVAL
*  FOR AVERAGED MONTHLY DATA
*
         DO 100 M = 1, 12
         IF ( FIRST .GT. NUMBER(M) .AND.
     *        FIRST .LE. NUMBER(M + 1) ) MFIRST = M
         IF ( LAST  .GT. NUMBER(M) .AND.
     *        LAST  .LE. NUMBER(M + 1) ) MLAST = M
 100     CONTINUE
 110     CONTINUE
*
*  HEAT EXCHANGE DATA
*
         READ ( JFILE, 520 ) CHECK, YEAR(2)
         IF ( CHECK .EQ. XSTOP ) GO TO 410
         IF ( CHECK .NE. XEQUI ) CALL ERROR ( 1330 )
         READ ( JFILE, 500 ) ( EQTEMP(N), N = FIRST, LAST )
```

```
      READ ( JFILE, 510 ) CHECK
      IF ( CHECK .NE. XEXCH ) CALL ERROR ( 1340 )
      READ ( JFILE, 500 ) ( EXCOEF(N), N = FIRST, LAST )
      READ ( JFILE, 510 ) CHECK
      IF ( CHECK .NE. XSHOR ) CALL ERROR ( 1350 )
      READ ( JFILE, 500 ) ( SOLAR (N), N = FIRST, LAST )
*
*  WIND SPEED DATA
*
      READ ( JFILE, 510 ) CHECK
      IF ( CHECK .NE. XWIND ) CALL ERROR ( 1355 )
      READ ( JFILE, 500 ) ( WIND(N), N = FIRST, LAST )
*
*  INFLOW DATA
*
      DO 270 L = 1, NIP
*
**  QUANTITY
*
      READ ( JFILE, 520 ) CHECK, YEAR(1), UNITS, TYPE
      IF ( CHECK .NE. XINFL ) CALL ERROR ( 1360 )
      IF ( UNITS .NE. XCFS  .AND.
     *     UNITS .NE. XKACF .AND.
     *     UNITS .NE. BLANK ) CALL ERROR ( 1370 )
      QINCFS = UNITS .NE. XKACF
      IF ( TYPE .EQ. XMONT ) GO TO 120
*
**  ENTIRE RECORD OF
**  INFLOW QUANTITIES
*
      READ ( JFILE, 500 ) ( INFLO(N, L), N = FIRST, LAST )
      GO TO 150
 120  CONTINUE
*
**  AVERAGED MONTHLY
**  INFLOW QUANTITIES
*
      READ ( JFILE, 500 ) ( VALUE(M), M = MFIRST, MLAST )
      DO 140 M = MFIRST, MLAST
      XYZ = VALUE(M)
      DFIRST = NUMBER(M) + 1
      DLAST  = NUMBER(M + 1)
      DO 130 N = DFIRST, DLAST
      INFLO(N, L) = XYZ
 130  CONTINUE
 140  CONTINUE
 150  CONTINUE
*
**  TEMPERATURE
      READ ( JFILE, 510 ) CHECK, UNITS, TYPE
      IF ( CHECK .NE. XTEMP ) CALL ERROR ( 1380 )
      IF ( UNITS .NE. XCELS .AND.
     *     UNITS .NE. XFAHR .AND.
     *     UNITS .NE. BLANK ) CALL ERROR ( 1390 )
      QINTC = UNITS .NE. XFAHR
      IF ( TYPE .EQ. XSINC ) GO TO 160
      IF ( TYPE .EQ. XMONT ) GO TO 180
*
**  ENTIRE RECORD OF
**  INFLOW TEMPERATURES
*
      READ ( JFILE, 500 ) ( INTEMP(N, L), N = FIRST, LAST )
```

```
        GO TO 210
 160    CONTINUE
*
**   HARMONIC GENERATION OF
**   INFLOW TEMPERATURES
*
        READ ( JFILE, 500 ) A, B, C, D
        DO 170 N = FIRST, LAST
        INTEMP(N, L) = A * SIN ( B * FLOAT ( N ) + C ) + D
 170    CONTINUE
        GO TO 210
 180    CONTINUE
*
**   AVERAGED MONTHLY
**   INFLOW TEMPERATURES
*
        READ ( JFILE, 500 ) ( VALUE(M), M = MFIRST, MLAST )
        DO 200 M = MFIRST, MLAST
        XYZ = VALUE(M)
        DFIRST = NUMBER(M) + 1
        DLAST  = NUMBER(M + 1)
        DO 190 N = DFIRST, DLAST
        INTEMP(N, L) = XYZ
 190    CONTINUE
 200    CONTINUE
 210    CONTINUE
*
**   QUALITITES
*
        IF ( QNKWAL ) GO TO 260
        DO 250 J = 1, NQUAL
        READ ( JFILE, 510 ) CHECK, UNITS, TYPE
        IF ( CHECK .NE. XQUAL ) CALL ERROR ( 1400 )
        IF ( TYPE .EQ. XMONT ) GO TO 220
*
**   ENTIRE RECORD OF
**   INFLOW QUALITIES
*
        READ ( JFILE, 500 ) ( INQUAL(N, L, J), N = FIRST, LAST )
        GO TO 250
 220    CONTINUE
*
**   AVERAGED MONTHLY
**   INFLOW QUALITIES
*
        READ ( JFILE, 500 ) ( VALUE(M), M = MFIRST, MLAST )
        DO 240 M = MFIRST, MLAST
        XYZ = VALUE(M)
        DFIRST = NUMBER(M) + 1
        DLAST  = NUMBER(M + 1)
        DO 230 N = DFIRST, DLAST
        INQUAL(N, L, J) = XYZ
 230    CONTINUE
 240    CONTINUE
 250    CONTINUE
 260    CONTINUE
 270    CONTINUE
*
*  OUTFLOW DATA
*
        IF ( QVERI ) GO TO 310
*
```

```
**   PREDICTION
*
      READ ( JFILE, 510 ) CHECK, UNITS, TYPE
      IF ( CHECK .NE. XOUTF ) CALL ERROR ( 1410 )
      IF ( UNITS .NE. XCFS  .AND.
     *     UNITS .NE. XKACF .AND.
     *     UNITS .NE. BLANK ) CALL ERROR ( 1420 )
      QOCFS = UNITS .NE. XKACF
      IF ( TYPE .EQ. XMONT ) GO TO 280
*
**   ENTIRE RANGE OF
**   OUTFLOW QUANTITIES
*
      READ ( JFILE, 500 ) ( SUMFLO(N), N = FIRST, LAST )
      GO TO 360
 280  CONTINUE
*
**   AVERAGED MONTHLY TOTAL
**   OUTFLOW QUANTITIES
*
      READ ( JFILE, 500 ) ( VALUE(M), M = MFIRST, MLAST )
      DO 300 M = MFIRST, MLAST
      XYZ = VALUE(M)
      DFIRST = NUMBER(M) + 1
      DLAST  = NUMBER(M + 1)
      DO 290 N = DFIRST, DLAST
      SUMFLO(N) = XYZ
 290  CONTINUE
 300  CONTINUE
      GO TO 400
 310  CONTINUE
*
**   VERIFICATION
*
      IF ( .NOT. QPORT ) GO TO 360
      DO 350 K = 1, NPORTS
      READ ( JFILE, 510 ) CHECK, UNITS, TYPE
      IF ( CHECK .NE. XOUTF ) CALL ERROR ( 1430 )
      IF ( UNITS .NE. XCFS  .AND.
     *     UNITS .NE. XKACF .AND.
     *     UNITS .NE. BLANK ) CALL ERROR ( 1440 )
      QOCFS = UNITS .NE. XKACF
      IF ( TYPE .EQ. XMONT ) GO TO 320
*
**   ENTIRE RANGE OF
**   OUTFLOW QUANTITIES
**   FOR EACH PORT
*
      READ ( JFILE, 500 ) ( OUTFLO(N, K), N = FIRST, LAST )
      GO TO 350
 320  CONTINUE
*
**   AVERAGED MONTHLY
**   OUTFLOW QUANTITIES
**
**   FOR EACH PORT
*
      READ ( JFILE, 500 ) ( VALUE(M), M = MFIRST, MLAST )
      DO 340 M = MFIRST, MLAST
      XYZ = VALUE(M)
      DFIRST = NUMBER(M) + 1
      DLAST  = NUMBER(M + 1)
      DO 330 N = DFIRST, DLAST
      OUTFLO(N, K) = XYZ
 330  CONTINUE
 340  CONTINUE
 350  CONTINUE
 360  CONTINUE
*
**  WEIR FLOWS
*
      IF ( .NOT. QWEIR ) GO TO 400
      READ ( JFILE, 510 ) CHECK, UNITS, TYPE
      IF ( CHECK .NE. XWEIR ) CALL ERROR ( 1450 )
      IF ( UNITS .NE. XCFS  .AND.
     *     UNITS .NE. XKACF .AND.
     *     UNITS .NE. BLANK ) CALL ERROR ( 1460 )
      IF ( TYPE .EQ. XMONT ) GO TO 370
*
**  ENTIRE RANGE OF
**  WEIR FLOWS
*
      READ ( JFILE, 500 ) ( WRFLO(N), N = FIRST, LAST )
      GO TO 400
 370  CONTINUE
*
**  AVERAGED MONTHLY
**  WEIR FLOWS
*
      READ ( JFILE, 500 ) ( VALUE(M), M = MFIRST, MLAST )
      DO 390 M = MFIRST, MLAST
      XYZ = VALUE(M)
      DFIRST = NUMBER(M) + 1
      DLAST  = NUMBER(M + 1)
      DO 380 N = DFIRST, DLAST
      WRFLO(N) = XYZ
 380  CONTINUE
 390  CONTINUE
 400  CONTINUE
      RETURN
 410  CONTINUE
      STOP
 500  FORMAT ( 8F10.0 )
 510  FORMAT ( A4, 46X, A4, 6X, A4 )
 520  FORMAT ( A4, 36X, I4, 2 ( 6X, A4 ) )
      END
      SUBROUTINE INFLOW
*
*  PLACE INFLOW QUANTITY AND QUALITY
*  INTO LAKE AT LEVEL OF INFLOW DENSITY
*  CALCULATE RE-DISTRIBUTION OF
*  LAYERS DUE TO INFLOW PROCESS
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
      COMMON / BB / TEMP(100), HGT(100), VOLUME(100), TARGET, TMIX, STAB
      COMMON / CC / DEN(100), NUSURF
      COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
      COMMON / EE / KFILE, LFILE, JFILE, IFILE
      COMMON / GG / FLOWIN(3), TEMPIN(3), QUALIN(3, 3)
      COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
      COMMON / II / EK, ET, SHORT, BETA, LAMDA, MIXCOEF, GAMMA,
     *              DECAY
      COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
      COMMON / UU / ENFLOW(100), SUMFLO(366), DPRINT(14)
      COMMON / WW / QNKWAL, YEAR(2)
      COMMON / XX / QJUNK, ONE, TWO
      REAL INFLO, INLAY, INFLOE
      INTEGER DAY
      LOGICAL QJUNK, QNKWAL
      DIMENSION UPQUAL(3), QMIX(3), QVMIX(3), INLAY(100)
      DATA C1, C2 / - 3.9863, 508929.2 /
      DATA C3, C4 / 288.9414, 68.12963 /
      DENFUN ( T ) = 1. - ( T + C1 ) ** 2 / C2
     *                 * ( T + C3 ) / ( T + C4 )
*
*  INITIALIZE FLOW INTO EACH LAYER
*
      DO 100 I = 1, LSURF
      ENFLOW(I) = 0.
 100  CONTINUE
      DO 330 L = 1, NIP
      INFLO = FLOWIN(L)
      IF ( INFLO .LE. 0. ) GO TO 330
*
*  CALCULATE MIXED TEMPERATURES AND
*  QUALITIES OF ENTRAINED VOLUME
*
*  USE ZERO ENTRAINMENT, GAMMA
*
      ENTFLO = GAMMA * INFLO
      SUMFL1 = INFLO + ENTFLO
      FLOW = ENTFLO
*
**  INITIALIZE MIXED QUALITIES
*
      TVMIX = 0.
      IF ( QNKWAL ) GO TO 120
      DO 110 J = 1, NQUAL
      QVMIX(J) = 0.
 110  CONTINUE
 120  CONTINUE
*
**  LOWER SURFACE TO ACCOUNT FOR ENTRAINMENT
*
      DO 180 I = 1, LSURF
      K = LSURF - I + 1
      VOLHGT = VOLUME(K) * HGT(K)
      IF ( FLOW .GE. VOLHGT ) GO TO 150
      TVMIX = TVMIX + TEMP(K) * FLOW
      ENFLOW(K) = ENFLOW(K) - FLOW
      IF ( QNKWAL ) GO TO 140
      DO 130 J = 1, NQUAL
      QVMIX(J) = QVMIX(J) + QUAL(J, K) * FLOW
 130  CONTINUE
 140  CONTINUE
      DEPTH = DELZ * ( FLOAT ( K - 1 ) + HGT(K) - FLOW / VOLUME(K) )
      HGT(K) = DEPTH / DELZ - FLOAT ( K - 1 )
      GO TO 190
 150  CONTINUE
      TVMIX = TVMIX + TEMP(K) * VOLHGT
      ENFLOW(K) = ENFLOW(K) - VOLHGT
      IF ( QNKWAL )  GO TO 170
      DO 160 J = 1, NQUAL
      QVMIX(J) = QVMIX(J) + QUAL(J, K) * VOLHGT
 160  CONTINUE
 170  CONTINUE
      HGT(K) = 0.
      FLOW = FLOW - VOLHGT
 180  CONTINUE
      WRITE(6,501)
 501  FORMAT(/30X,'WARNING ENTRAINED INFLOW = POOL VOLUME')
      ENTFLO = ENTFLO - FLOW
      SUMFL1 = INFLO + ENTFLO
 190  CONTINUE
      LSURF = K
*
*  CALCULATE MIXED QUALITIES OF
*  ENTRAINED VOLUME AND INFLOW QUANTITY
*
      TVMIX = TVMIX + INFLO * TEMPIN(L)
      IF ( QNKWAL ) GO TO 210
      DO 200 J = 1, NQUAL
      QVMIX(J) = QVMIX(J) + INFLO * QUALIN(L, J)
 200  CONTINUE
 210  CONTINUE
      TMIX = TVMIX / SUMFL1
      IF ( QNKWAL ) GO TO 230
      DO 220 J = 1, NQUAL
      QMIX(J) = QVMIX(J) / SUMFL1
 220  CONTINUE
 230  CONTINUE
*
*  CALCULATE DENSITY OF INFLOW
*
      DENST = DENFUN ( TMIX )
      DENSQ = 0.
      IF ( QNKWAL ) GO TO 250
      DO 240 J = 1, NQUAL
      DENSQ = DENSQ + QMIX(J) * DENC(J)
 240  CONTINUE
 250  CONTINUE
      DENMIX = DENST + DENSQ
*
*  DETERMINE LAYER OF INFLOW
*
      DO 260 I = 1, LSURF
      INFLAY = I
      IF ( DENMIX .GT. DEN(I) ) GO TO 270
 260  CONTINUE
      INFLAY = LSURF
 270     CONTINUE
*
*  INITIALIZE INPUT INTO INFLOW LAYER
*
         UPFLOW = 0.
         DO 280 I = 1, MAXLAY
         INLAY(I) = 0.
 280     CONTINUE
         INLAY(INFLAY) = SUMFL1
         ENFLOW(INFLAY) = ENFLOW(INFLAY) + SUMFL1
*
*  COMPUTE NEW TEMPERATURES AND QUALITY
*  VALUES FOR LAYERS ABOVE INFLOW LAYER
*
         I = INFLAY - 1
 290     CONTINUE
         I = I + 1
         IF ( I .GT. MAXLAY ) CALL ERROR ( 2020 )
         INFLOE = INLAY(I)
         VOLHGT = VOLUME(I) * HGT(I)
         SUMVOL = UPFLOW + INFLOE + VOLHGT
         TEMP(I) = ( UPFLOW * UPTEMP + VOLHGT * TEMP(I) +
     *                INFLOE * TMIX ) / SUMVOL
         UPTEMP = TEMP(I)
         IF ( QNKWAL ) GO TO 310
         DO 300 J = 1, NQUAL
         QUAL(J, I) = ( UPFLOW * UPQUAL(J) + VOLHGT * QUAL(J, I) +
     *                   INFLOE * QMIX(J) ) / SUMVOL
         UPQUAL(J) = QUAL(J, I)
 300     CONTINUE
 310     CONTINUE
         UPVOL = UPFLOW + INFLOE + VOLHGT - VOLUME(I)
         IF ( UPVOL .LE. 0. ) GO TO 320
         UPFLOW = UPVOL
         HGT(I) = 1.
         GO TO 290
 320     CONTINUE
*
*  CALCULATE NEW WATER SURFACE AFTER INFLOW
*
         LSURF = I
         HGT(LSURF) = 1. + UPVOL / VOLUME(LSURF)
 330     CONTINUE
         DEPTH = DELZ * ( FLOAT ( LSURF - 1 ) + HGT(LSURF) )
         IF ( QJUNK ) WRITE ( KFILE, 500 ) DAY
         IF ( QJUNK ) WRITE ( KFILE, 510 )
     *                        ( I, TEMP(I), I = 1, LSURF )
         IF ( QJUNK ) WRITE ( KFILE, 520 ) LSURF, HGT(LSURF)
         RETURN
 500     FORMAT ( /// 5X, 'DAY - ', I3, 5X, 'AFTER INFLOW' )
 510     FORMAT ( /// ( 8 ( 5X, I2, 2X, F6.2 ) ) )
 520     FORMAT ( 5X, 'HGT(', I2, ') = ', F5.3 )
         END
```

```
        SUBROUTINE LINPLT
*
*  THIS SUBROUTINE PLOTS PROFILES OF TEMPERATURE
*  AND AN ARBITRARY NUMBER OF QUALITY PARAMETERS
*
        COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
        COMMON / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
        COMMON / CH / NM, TITLE
        COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
        COMMON / EE / KFILE, LFILE, JFILE, IFILE
        COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
        COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
        COMMON / PP / FIRST, LAST, NJ, INDEX(366)
        COMMON / WW / QNKWAL, YEAR(2)
        DIMENSION P(3), MAXQ(3), CONCOL(3), TSPACE(11), CSPACE(3, 11)
        CHARACTER*1 T, P, PEGGED, X, BLANK, PLUS, CFIRST, CLAST
        CHARACTER TITLE*78, COLUMN(100)*1, NM*3
        INTEGER YEAR, DAY, TMPCOL, PORT, OPEN, TSPACE, CSPACE, CHANGE
        LOGICAL QRANGE, QNKWAL
        DATA PEGGED, T, P / '*', 'T', '1', '2', '3' /
        DATA X, BLANK, PLUS / 'X', ' ', '+' /
        DATA MAXT / 50 /
        DATA MAXQ / 20, 10,  500 /
*
*  PRINT DATE INFORMATION
*
        WRITE ( LFILE, 500 ) TITLE
        IF ( YEAR(1) .GT. 0 ) WRITE ( LFILE, 510 ) YEAR
        WRITE ( LFILE, 520 ) DAY, NJ, NM
*
*  DETERMINE TEMPERATURE AXIS SPACING
*
        CHANGE = MAXT / 10
        TSPACE(1) = 0
        DO 100 K = 1, 10
        TSPACE(K + 1) = TSPACE(K) + CHANGE
 100    CONTINUE
*
*  DETERMINE CONCENTRATION AXES SPACING
*
        IF ( QNKWAL ) GO TO 130
        DO 120 J = 1, NQUAL
        CHANGE = MAXQ(J) / 10
        CSPACE(J, 1) = 0
        DO 110 K = 1, 10
        CSPACE(J, K + 1) = CSPACE(J, K) + CHANGE
 110    CONTINUE
 120    CONTINUE
 130    CONTINUE
*
*  PRINT TEMPERATURE BANNER AND AXIS
*
        WRITE ( LFILE, 530 )
        WRITE ( LFILE, 540 ) ( TSPACE(K), K = 1, 11 )
        WRITE ( LFILE, 550 )
*
*  BEGIN TO FILL IN COLUMN ARRAY
*
        ELEV = BOTTOM + DEPTH + DELZ
        DEEP = - DELZ
        DO 220 I = 1, LSURF
        K = LSURF - I + 1
```

```
         DEEP = DEEP + DELZ
         ELEV = ELEV - DELZ
*
*  BLANK OUT COLUMN ARRAY
*
         CFIRST = BLANK
         CLAST = BLANK
         DO 140 L = 1, 100
         COLUMN(L) = BLANK
 140     CONTINUE
*
*  DETERMINE IF TEMPERATURE AND QUALITY
*  VALUES ARE WITHIN RANGE OF PLOT
*
         TMPCOL = 1. + TEMP(K) * 100. / FLOAT ( MAXT )
         IF ( QNKWAL ) GO TO 160
         DO 150 J = 1, NQUAL
         CONCOL(J) = 1. + QUAL(J, K) * 100. / FLOAT ( MAXQ(J) )
 150     CONTINUE
 160     CONTINUE
         IF ( TMPCOL .LT. 0 ) CFIRST = PEGGED
         IF ( TMPCOL .GT. 100 ) CLAST = PEGGED
         IF ( QNKWAL ) GO TO 180
         DO 170 J = 1, NQUAL
         IF ( CONCOL(J) .GT. 100 ) CLAST = PEGGED
         IF ( CONCOL(J) .LT. 0. ) CFIRST = PEGGED
 170     CONTINUE
 180     CONTINUE
*
*  DETERMINE COLUMN FOR
*  PLOTTING EACH COMPONENT
*
         QRANGE = TMPCOL .GE. 0 .AND. TMPCOL .LE. 100
         IF ( .NOT. QRANGE ) GO TO 190
         IJK = TMPCOL
         COLUMN(IJK) = T
 190     CONTINUE
         IF ( QNKWAL ) GO TO 210
         DO 200 J = 1, NQUAL
         QRANGE = CONCOL(J) .GE. 0 .AND. CONCOL(J) .LE. 100
         IF ( .NOT. QRANGE ) GO TO 200
         IJK = CONCOL(J)
         IF ( COLUMN(IJK) .EQ. BLANK ) COLUMN(IJK) = P(J)
         IF ( COLUMN(IJK) .NE. BLANK ) COLUMN(IJK) = X
 200     CONTINUE
 210     CONTINUE
*
*  PRINT ONE LINE OF PLOT
*
         WRITE ( LFILE, 560 ) ELEV, DEEP, PLUS, CFIRST, COLUMN, CLAST
 220     CONTINUE
*
*  PRINT BOTTOM AXES
*
         IF ( QNKWAL ) GO TO 240
         DO 230 J = 1, NQUAL
         WRITE ( LFILE, 550 )
         WRITE ( LFILE, 540 ) ( CSPACE(J, L), L = 1, 11 )
         WRITE ( LFILE, 570 ) J
 230     CONTINUE
         GO TO 250
 240     CONTINUE
```

```
          WRITE ( LFILE, 550 )
          WRITE ( LFILE, 540 ) ( TSPACE(K), K = 1, 11 )
          WRITE ( LFILE, 530 )
 250      CONTINUE
*
*  QUIT
*
          RETURN
 500      FORMAT ( '1' // 20X, A )
 510      FORMAT ( 20X, I4, 'H - HYDROLOGY' /
     *             20X, I4, 'H - METEOROLOGY' )
 520      FORMAT ( // 30X, 'DAY - ', I3,
     *             5X, I2, 1X, A3 /// )
 530      FORMAT ( 30X, 'TEMPERATURE  ( DEG-C )'  / )
 540      FORMAT ( 11X, 11 I10 )
 550      FORMAT ( 20X, 10 ( 'I---------' ), 'I' )
 560      FORMAT ( 1X, F5.0, 6X, F5.0, A2, 102A1 )
 570      FORMAT ( 30X, 'QUALITY -', 1X, I1, 3X, '( MG/L )' // )
          END
```

```
        SUBROUTINE MIXING
*
*  INTERNAL MIXING OF LAKE TEMPERATURE
*  USING INTEGRAL WIND AND CONVECTIVE
*  ENERGY IN THE EPILIMNION AND EDDY
*  DIFFUSION IN THE HYPOLIMNION
*  BASED ON WORK OF FORD AND HARLEMAN & BLOSS.
*
        COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
        COMMON / BB / TEMP(100), HGT(100), VOLUME(100), TARGET, TMIX, STAB
        COMMON / CC / DEN(100). NUSURF
        COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
        COMMON / EE / KFILE, LFILE, JFILE, IFILE
        COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
        COMMON / II / EK, ET, SHORT, BETA, LAMDA, MIXCOEF, GAMMA,
     *                DECAY
        COMMON / JJ / SUMOUT, WIDTH(100)
        COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
        COMMON / XX / QJUNK, ONE, TWO
        COMMON / BL / WIND(366)
        COMMON / WW / QNKWAL, YEAR(2)
        DIMENSION AREA(100), DELT(100), DELQ(3,100), QMIX(3)
        INTEGER DAY, TLAYER
        LOGICAL QFIRST, QJUNK, QHELP, QNKWAL
        REAL MIXCOEF
        NAMELIST / M1NAME / DAY, AIRRHO, CDRAG, WSPEED,
     *                      SHEAR, LSURF, DEN, C10, WSTAR
        NAMELIST / M2NAME / TKEWIND, TKEVECT, WORK, DELRHO, TKE,
     *                      DELZONE, ZGRAVTY, EK, ET,
     *                      TKECONV,DISSIP,RI,FRI,SUMWORK,
     *                      RHOMIX,SUMVOL,ZONE
        NAMELIST / M3NAME / TKEWIND, TKECONV, SUMWORK,
     *                      DISSIP, M, TLAYER
        DATA QFIRST / .TRUE. /
        DATA CDRAG, CCOEF, WCOEF / 1.3E-03, .05, 0.5 /
        DATA AIRRHO / 1.2 /
        DATA MIXCOEF / 1. /
        DATA EXPAND, SPHEAT, G / 2.E - 04, 1. , 9.806 /
        DATA C1, C2 / - 3.9863, 508929.2 /
        DATA C3, C4 / 288.9414, 68.12963 /
        DENFUN ( T ) = 1. - ( T + C1 ) ** 2 / C2 *
     *                      ( T + C3 ) / ( T + C4 )
*
*  COMPUTE THE HORIZONTAL AREAS AND THE
*  FIRST DERIVATIVES OF THE AREAS
*
        QHELP = .TRUE.
        IF ( .NOT. QFIRST ) GO TO 110
        QFIRST = .FALSE.
        DO 100 I = 1, MAXLAY
        AREA(I) = VOLUME(I) / DELZ
 100    CONTINUE
        TLAYER = LSURF
 110    CONTINUE
*
*  COMPUTE DENSITY PROFILE
*
        DO 120 I = 1, LSURF
        DENSQ = 0.
        IF ( QNKWAL ) GO TO 118
        DO 115 J = 1, NQUAL
        DENSQ = DENSQ + QUAL(J,I) * DENC(J)
```

```
  115     CONTINUE
  118     CONTINUE
          DEN(I) = DENFUN ( TEMP(I) ) + DENSQ
  120     CONTINUE
*
*  CHECK FOR STABLE DENSITY PROFILE
*
          CALL STABLE
*
*  SOLVE FOR DRAG COEF USING SUCCESIVE APPROX
*
          IF ( WIND(DAY) .LT. 1.0 ) C10 = 5.31E-04
          IF ( WIND(DAY) .LT. 1.0 ) GO TO 126
          WSP = .447 * WIND(DAY)
          C10 = CDRAG
          DO 125 I = 1,3
          C10 = 1. / ( 2.38 * ALOG( 8915. / ( C10 *
     *       WSP ** 2. ) ) ) ** 2.
  125     CONTINUE
  126     CONTINUE
*
*  COMPUTE SHEAR STRESS
*  AND SHEAR VELOCITY
*
          SHEAR = AIRRHO * C10 * WIND(DAY) ** 2
          WSPEED = WIND(DAY)
          WSTAR = SQRT ( SHEAR / DEN(LSURF) )
          WSTAR = AMAX1 ( 1.0E-05, WSTAR )
          IF ( QHELP .AND. QJUNK ) WRITE ( LFILE, M1NAME )
*
*  COMPUTE TURBULENT KINETIC
*  ENERGY FROM WIND SHEAR
*
          CWD = WCOEF
          TKEWIND = AREA(TLAYER) * WSTAR * SHEAR * CWD
*
*   CONVERT KINETIC ENERGY TO JOULES
*
          TKEWIND = TKEWIND * 9.8756E + 08
*
*  INITIALIZE SUMMATIONS
*
          SUMVOL = VOLUME(LSURF) * HGT(LSURF)
          ZGRAVTY = .5 * DELZ * HGT(LSURF)
          RHOMIX = DEN(LSURF)
          TEMPMIX = TEMP(LSURF)
          IF ( QNKWAL ) GO TO 134
          DO 133 J = 1, NQUAL
          QMIX(J) = QUAL(J,LSURF)
  133     CONTINUE
  134     CONTINUE
          ZONE = DELZ * HGT(LSURF)
          SUMWORK = 0.
          DISSIP = 0.
*
*  BEGIN SEARCH FOR LOWER
*  LAYER OF WIND-MIXED ZONE
*
          LSM = LSURF - 1
          DO 130 I = 1, LSM
          M = LSURF - I
*
```

```
*  COMPUTE TURBULENT KINETIC ENERGY
*  DUE TO OVERTURNING CONVECTION
*
*  NET HEAT - (BTU/SQ FT/DAY)
*  SPECIFIC HEAT - ( CAL/DEG-F /G )
*  DENSITY OF AIR - (G/L)
*
        THETA = 9. / 5. * TEMP(LSURF) + 32.
        HEAT = - EK * ( ET - THETA )
        TKECONV = CCOEF * HEAT * AREA(LSURF) *
     *            ZONE * EXPAND / SPHEAT * G
        TKEVECT = TKECONV
        TKECONV = AMAX1 ( 0., TKECONV )
        TKECONV = 0.
*
*   CONVERT KINETIC ENERGY TO JOULES
*
        TKECONV = TKECONV * 3.3458E + 06
*
*   FIND TOTAL TURBULENT KINETIC ENERGY
*
        TKE = TKEWIND + TKECONV - SUMWORK - DISSIP
*
*
*  COMPUTE WORK REQUIRED TO LIFT
*  THE CURRENT LAYER -M- TO THE CENTER
*  OF MASS OF THE NEW MIXED ZONE
*
        DELRHO = DEN(M) - RHOMIX
        DELRHO = AMAX1 ( 0., DELRHO )
        DELVOL = VOLUME(M)
        DELZONE = ZONE + 0.5 * DELZ
        ZGRAVTY = ( ZGRAVTY * SUMVOL + DELZONE * DELVOL ) /
     *            ( SUMVOL + DELVOL )
        WORK = DELRHO * DELVOL* G * ( DELZONE - ZGRAVTY )
*
*  CONVERT WORK TO JOULES
*
        WORK = WORK * 3.76E + 08
*
*  COMPUTE RICHARDSON NO.
*  FOR DISSIPATION
*
        RI = ( G * DELRHO * ZONE ) / WSTAR ** 2.
        RI = 1525.18 * RI
        IF ( DELRHO .LT. 1.0E-05 ) GO TO 127
        FRI = .057 * RI * ( 29.5 - RI ** .5 )
     *      / ( 14.2 + RI )
        FRI = AMIN1 ( FRI, 1.0 )
        FRI = AMAX1 ( FRI, 0. )
        DISSIP = ( 1. - FRI ) * TKE
        GO TO 128
 127    CONTINUE
        DISSIP = 0.
 128    CONTINUE
*
*  COMPARE WORK AND TOTAL
*  TURBULENT KINETIC ENERGY
*
        IF ( QHELP .AND. QJUNK ) WRITE ( LFILE, M2NAME )
        IF ( WORK .GT. TKE ) GO TO 140
*
```

```
*  UPDATE MIXED ZONE, SUMWORK, AND DISSIPATION
*
        RHOMIX = ( RHOMIX * SUMVOL + DEN(M) * DELVOL ) /
     *           ( SUMVOL + DELVOL )
        TEMPMIX = ( TEMPMIX * SUMVOL + TEMP(M) * DELVOL ) /
     *            ( SUMVOL + DELVOL )
        IF ( QNKWAL ) GO TO 740
        DO 730 J = 1, NQUAL
        QMIX(J)  =  ( QMIX(J) * SUMVOL + QUAL(J,M) * DELVOL ) /
     *            ( SUMVOL + DELVOL )
 730    CONTINUE
 740    CONTINUE
        SUMVOL = SUMVOL + DELVOL
        ZONE = ZONE + DELZ
        SUMWORK = SUMWORK + WORK
 130    CONTINUE
 140    CONTINUE
        TLAYER = M
*
*  COMPUTE EDDY DIFFUSION COEFFICIENTS
*  FOR THE HYPOLIMNION - FUTURE ENHANCEMENT
*
*
*  COMPUTE THE CHANGES IN THE
*  TEMPERATURE PROFILES FOR TLAYER AND BELOW
*  USING EDDY DIFFUSION
*
        LSM1 = TLAYER
        DO 170 I = 2, LSM1
        CMIX = MIXCOEF * EXP(-DECAY*(DEN(I+1)-DEN(I-1))**2.)
        DELT(I) = CMIX / ( AREA(I) * 2. * DELZ ** 2 ) *
     *          ( ( AREA(I + 1) + AREA(I) ) *
     *            ( TEMP(I + 1) - TEMP(I) ) -
     *            ( AREA(I) + AREA(I - 1) ) *
     *            ( TEMP(I) - TEMP(I - 1) ) )
 170    CONTINUE
      DELT(1) = MIXCOEF * (AREA(2)+AREA(1)) *
     *          (TEMP(2)-TEMP(1)) / (AREA(1)*DELZ*DELZ)
*
*  COMPUTE CHANGES IN QUALITY PROFILES FOR TLAYER AND BELOW
*
        IF ( QNKWAL ) GO TO 450
             DO 430 I = 2, LSM1
        DO 420 J = 1, NQUAL
        CMIX = MIXCOEF * EXP(-DECAY*(DEN(i+1)-DEN(I-1))**2.)
        DELQ(J,I) = CMIX / ( AREA(I) * 2. * DELZ ** 2 ) *
     *          ( ( AREA(I + 1) + AREA(I) ) *
     *            ( QUAL(J,I+1) - QUAL(J,I) ) +
     *            ( AREA(I + 1) - AREA(I) ) *
     *            ( QUAL(J,I+1) + QUAL(J,I) ) -
     *            ( AREA(I) + AREA(I - 1) ) *
     *            ( QUAL(J,I) - QUAL(J,I-1) ) -
     *            ( AREA(I) - AREA(I - 1) ) *
     *            ( QUAL(J,I) + QUAL(J,I-1) ) )
        IF ( ABS(DELQ(J,I)) .GT. .5*QUAL(J,I)) THEN
        DELQ(J,I) = SIGN(.5*QUAL(J,I),DELQ(J,I))
        END IF
 420    CONTINUE
 430    CONTINUE
        DO 440 J = 1, NQUAL
        DELQ(J,1)=MIXCOEF*(AREA(2)+AREA(1))*(QUAL(J,2)-QUAL(J,1))
     *            /(AREA(1)*DELZ*DELZ)
 440    CONTINUE
 450    CONTINUE
*
*  COMPUTE THE NEW TEMPERATURE PROFILE
*
        DO 180 I = 1, LSURF
        IF ( I .LE. TLAYER ) TEMP(I) = TEMP(I) + DELT(I)
        IF ( I .GT. TLAYER ) TEMP(I) = TEMPMIX
 180    CONTINUE
*
*  COMPUTE NEW QUALITY PROFILES
*
        IF ( QNKWAL ) GO TO 650
        DO 630 I = 1, LSURF
        DO 620 J = 1, NQUAL
        IF ( I .LE. TLAYER ) QUAL(J,I) = QUAL(J,I) + DELQ(J,I)
        IF ( I .GT. TLAYER ) QUAL(J,I) = QMIX(J)
 620    CONTINUE
 630    CONTINUE
 650    CONTINUE
*
*  COMPUTE NEW DENSITIES
*  AFTER DIFFUSION PROCESS
*
        DO 190 I = 1, LSURF
        DENSQ = 0.
        IF ( QNKWAL ) GO TO 186
        DO 183 J = 1, NQUAL
        DENSQ = DENSQ + QUAL(J,I) * DENC(J)
 183    CONTINUE
 186    CONTINUE
        DEN(I) = DENTUN ( TEMP(I) ) + DENSQ
 190    CONTINUE
        IF ( QHELP .AND. QJUNK ) WRITE ( KFILE, M3NAME )
        IF ( .NOT. QJUNK ) RETURN
        WRITE ( KFILE, 510 ) DAY, TLAYER, TEMPMIX
        WRITE ( KFILE, 500 ) ( I, TEMP(I), I = 1, LSURF )
        RETURN
 500    FORMAT ( /// ( 8 ( 5X, I2, 2X, F6.2 ) ) )
 510    FORMAT ( /// 5X, 'DAY - ', I3, 5X, 'AFTER MIXING' ,I3,E12.4)
        END
```

```
      SUBROUTINE OUTVEL
*
*  SET UP SELECTIVE WITHDRAWAL PARAMETERS
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
     *       / BB / TEMP(100), HGT(100), VOLUME(100), TARGET, TMIX, STAB
     *       / CC / DEN(100), NUSURF
     *       / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
     *       / FF / NPORTS, PAREA(8), PHGT(8)
     *       / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
     *       / JJ / SUMOUT, WIDTH(100)
     *       / MM / DAY, PORT(8), PHLOW(8), OPEN
     *       / XX / QJUNK, ONE, TWO
     *       / CD / QPORT, QWEIR, QVINCR, QINITC
     *       / DE / TOPLIM, LOWLIM, V(100), FLORAT, LL(8), LT(8)
     *       / EF / AREA, HGTPRT, LAYPRT, DENPRT
     *       / GH / WRFLOW, WRFLO(366)
     *       / IJ / WANGLE, WTHETA(8)
      LOGICAL QJUNK,   QPORT, QWEIR
      INTEGER PORT, OPEN, TOPLIM, DAY
      LAYER ( X )  = 1. + X / DELZ
*
*  INITIALIZE THE TOTAL VELOCITY PROFILE
*
      DO 100 I = 1, LSURF
      VEL(I) = 0
 100  CONTINUE
*
*  CALCULATE NEW DEPTH AFTER WITHDRAWAL
*
      FVOL  = SUMOUT
      DO 120 I = 1, LSURF
      K = LSURF - I + 1
      VOLHGT = VOLUME(K) * HGT(K)
      IF ( FVOL .GT. VOLHGT ) GO TO 110
      NUSURF = K
      DEPTH = DELZ * ( FLOAT ( NUSURF - 1 ) + HGT(K) -
     *                 FVOL / VOLUME(K) )
      GO TO 130
 110  CONTINUE
      FVOL = FVOL - VOLHGT
 120  CONTINUE
 130  CONTINUE
*
*  DETERMINE OUTFLOW
*  VELOCITY FROM PORTS
*
      IF ( .NOT. QPORT ) GO TO 160
      DO 150 K = 1, OPEN
      FLORAT = PHLOW(K)
      IF ( FLORAT .LE. 0. ) GO TO 150
      IJK    = PORT(K)
      AREA   = PAREA(IJK)
      HGTPRT = PHGT(IJK)
      LAYPRT = LAYER ( HGTPRT )
      WANGLE = WTHETA(IJK)
      CALL VPORT
      LL(K)  = LOWLIM
      LT(K)  = TOPLIM
      SUMVW  = 0.
      DO 135 I = LOWLIM, TOPLIM
        SUMVW  = SUMVW + V(I)
 135      CONTINUE
          SCALE  = FLORAT / SUMVW
*
**  ADD LOCAL VELOCITY
**  TO TOTAL PROFILE
*
          DO 140 I = LOWLIM, TOPLIM
          VEL(I) = VEL(I) + V(I) * SCALE
 140      CONTINUE
 150      CONTINUE
 160      CONTINUE
*
*  DETERMINE OUTFLOW
*  VELOCITY FROM WEIR
*
          IF ( .NOT. QWEIR ) GO TO 180
          FLORAT = WRFLOW
          IF ( FLORAT .LE. 0. ) GO TO 180
          CALL VWEIR
*
**  ADD LOCAL VELOCITY
**  TO TOTAL PROFILE
*
          SUMVW = 0.
          DO 165 I = LOWLIM, TOPLIM
          SUMVW = SUMVW + V(I)
 165      CONTINUE
          SCALE = FLORAT / SUMVW
          DO 170 I = LOWLIM, TOPLIM
          VEL(I) = VEL(I) + V(I) * SCALE
 170      CONTINUE
 180      CONTINUE
*
*  SCALE VELOCITY PROFILE
*
          VMAX = 0
          DO 190 I = 1, LSURF
          VMAX = AMAX1 ( VMAX, VEL(I) )
 190      CONTINUE
          IF ( VMAX .LE. 0. ) GO TO 210
          DO 200 I = 1, LSURF
          VEL(I) = VEL(I) / VMAX
 200      CONTINUE
 210      CONTINUE
          RETURN
          END
```

```
      SUBROUTINE REFILL
*
* CALCULATE WITHDRAWAL FROM EACH LAYER,
* DETERMINE RELEASE TEMPERATURE AND
* QUALITIES, AND REFILL EACH LAYER
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
      COMMON / BB / TEMP(100), HGT(100), VOLUME(100), TARGET, TMIX,
     *              STAB
      COMMON / CC / DEN(100), NUSURF
      COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
      COMMON / EE / KFILE, LFILE, JFILE, IFILE
      COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIF, WTHDRW(100)
      COMMON / JJ / SUMOUT, WIDTH(100)
      COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
      COMMON / OO / AVTEMP, AVQUAL(3), AVGT(366), AVGQ(366, 3)
      COMMON / WW / QNKWAL, YEAR(2)
      COMMON / XX / QJUNK, ONE, TWO
      DIMENSION SUMQF(3)
      INTEGER DAY
      LOGICAL QJUNK, QMOVE, QNKWAL
      DATA C1, C2 / - 3.9863, 508929.2 /
      DATA C3, C4 / 288.9414, 68.12963 /
      DENFUN ( T ) = 1. - ( T + C1 ) ** 2 / C2
     *                * ( T + C3 ) / ( T + C4 )
*
* COMPUTE SHAPE OF WITHDRAWAL PROFILE
* FROM VELOCITY AND WIDTH OF EACH LAYER
*
      SUM = 0.
      SUMF = 0
      TFLOW = 0.
      DO 100 I = 1, LSURF
         WTHDRW(I) = VEL(I) * HGT(I)
         SUM = SUM + WTHDRW(I)
 100  CONTINUE
      IF ( SUM .LE. 1.E - 10 ) GO TO 330
      SCALE = SUMOUT / SUM
      DO 110 I = 1, LSURF
         WTHDRW(I) = WTHDRW(I) * SCALE
         TFLOW = TFLOW + WTHDRW(I)
 110  CONTINUE
*
* DO NOT WITHDRAW WATER FROM LAYERS THAT WILL DISAPPEAR
* AFTER WITHDRAWAL FOR THIS SIMULATION DAY.
*
      IF ( LSURF .GT. NUSURF ) THEN
      NUSP1 = NUSURF + 1
      DO 129 I = NUSP1 , LSURF
         WTHDRW(NUSURF) = WTHDRW(NUSURF) + WTHDRW(I)
         WTHDRW(I) = 0.0
 129  CONTINUE
      END IF
*
* MAXIMUM WITHDRAWAL FROM ANY LAYER IS THE VOLUME OF THE LAYER.
* ANY EXTRA IS TAKEN FROM THE LAYER ABOVE.
*
      TFLOW = 0.
      EXTRA = 0.
      DO 131 I = 1, LSURF
         VOLHGT = VOLUME(I) * HGT(I)
         FLOW = WTHDRW(I) + EXTRA
```

```
            IF ( FLOW .LT. VOLHGT ) GO TO 120
            WTHDRW(I) = VOLHGT
            EXTRA = FLOW - VOLHGT
            GO TO 130
 120        CONTINUE
            WTHDRW(I) = FLOW
            EXTRA = 0.
 130        TFLOW = TFLOW + WTHDRW(I)
 131     CONTINUE
*
*  IF EXTRA FLOW REMAINS UNACCOUNTED
*  FOR THEN WITHDRAW IT FROM THE
*  HIGHEST POSSIBLE LAYERS
*
         IF ( EXTRA .LE. 0. ) GO TO 160
         DO 150 J = 1, LSURF
            K = LSURF - I + 1
            VOLHGT = VOLUME(K) * HGT(K)
            REMVOL = VOLHGT - WTHDRW(K)
            IF ( REMVOL .GE. EXTRA ) GO TO 140
            WTHDRW(K) = VOLHGT
            EXTRA = EXTRA - REMVOL
            GO TO 150
 140        CONTINUE
            WTHDRW(K) = WTHDRW(K) + EXTRA
            GO TO 160
 150     CONTINUE
 160     CONTINUE
*
*  INITIALIZE OUTLOW SUMMATIONS
*
         SUMF = 0.
         SUMTF = 0.
         IF ( QNKWAL ) GO TO 180
         DO 170 J = 1, NQUAL
            SUMQF(J) = 0.
 170     CONTINUE
 180     CONTINUE
*
*  WITHDRAW FLOW FROM LAYERS
*
         I = 0
 181     I = I + 1
         IF ( I .GE. LSURF ) GO TO 250
         FLOW = WTHDRW(I)
         IF ( FLOW .LE. 0. ) GO TO 181
*
*  SUM OUTFLOW CHARACTERISTICS
*
         SUMF = SUMF + FLOW
         SUMTF = SUMTF + TEMP(I) * FLOW
         IF ( QNKWAL ) GO TO 200
         DO 190 J = 1, NQUAL
            SUMQF(J) = SUMQF(J) + QUAL(J, I) * FLOW
 190     CONTINUE
 200     CONTINUE
*
*  CONSECUTIVELY REFILL LAYERS FROM ABOVE
*
         ISM = LSURF - 1
         DO 220 K = I, ISM
            VOL = VOLUME(K)
            REMVOL = VOL - FLOW
            VOLHGT = VOLUME(K + 1) * HGT(K + 1)
            DOWN = FLOW
            QMOVE = FLOW .GT. VOLHGT
            IF ( QMOVE ) DOWN = VOLHGT
            TREMV = TEMP(K) * REMVOL
            TDOWN = TEMP(K + 1) * DOWN
            VOL = REMVOL + DOWN
            TEMP(K) = ( TREMV + TDOWN ) / VOL
            IF ( QNKWAL ) GO TO 220
            DO 210 J = 1, NQUAL
               QREMV = QUAL(J, K) * REMVOL
               QDOWN = QUAL(J, K + 1) * DOWN
               QUAL(J, K) = ( QREMV + QDOWN ) / VOL
 210        CONTINUE
 220     CONTINUE
*
*  ADJUST WATER SURFACE TO ACCOUNT
*  FOR WITHDRAWAL FROM ONE LAYER
*
         IF ( QMOVE ) GO TO 230
         HGT(LSURF) = ( VOLUME(LSURF) * HGT(LSURF) -
     *               DOWN ) / VOLUME(LSURF)
         GO TO 181
 230     CONTINUE
         HGT(LSURF) = 0.
         LSURF = LSURF - 1
         HGT(LSURF) = ( REMVOL + DOWN ) / VOLUME(LSURF)
         WTHDRW(LSURF) = WTHDRW(LSURF) + WTHDRW(LSURF+1)
         IF ( I .EQ. LSURF ) THEN
         FLOW = WTHDRW(LSURF+1)
         WTHDRW(LSURF+1) = 0.0
         GO TO 252
         END IF
         WTHDRW(LSURF+1) = 0.0
         GO TO 181
 250     CONTINUE
*
*  ADJUST WATER SURFACE FOR
*  WITHDRAWAL FROM SURFACE LAYER
*
         FLOW = WTHDRW(LSURF)
         IF ( FLOW .LE. 0.0 ) GO TO 270
 252     VOLHGT = VOLUME(LSURF) * HGT(LSURF)
         REMVOL = VOLHGT - FLOW
         IF ( REMVOL .LT. 0.0 ) THEN
         FLOW = VOLHGT
         SUMF = SUMF + FLOW
         SUMTF = SUMTF + TEMP(LSURF) * FLOW
         HGT(LSURF) = 0.0
         LSURF = LSURF - 1
         FLOW = - REMVOL
         GO TO 252
         END IF
         HGT(LSURF) = REMVOL / VOLUME(LSURF)
         SUMF = SUMF + FLOW
         SUMTF = SUMTF + TEMP(LSURF) * FLOW
         IF ( QNKWAL ) GO TO 270
         DO 260 J = 1, NQUAL
            SUMQF(J) = SUMQF(J) + QUAL(J, LSURF) * FLOW
 260     CONTINUE
 270     CONTINUE
```

```
      ODEPTH = DEPTH
      DEPTH = DELZ * ( FLOAT ( LSURF - 1 ) + HGT(LSURF) )
*
*  CALCULATE AVERAGE RELEASE CHARACTERISTICS
*
      IF ( SUMF .LE. 0. ) GO TO 290
      AVTEMP = SUMTF / SUMF
      IF ( QNKWAL ) GO TO 290
      DO 280 J = 1, NQUAL
         AVQUAL(J) = SUMQF(J) / SUMF
 280  CONTINUE
 290  CONTINUE
*
*  COMPUTE RESULTANT DENSITY PROFILE
*
      DO 320 I = 1, LSURF
         DENST = DENFUN ( TEMP(I) )
         DENSQ = 0.
         IF ( QNKWAL ) GO TO 310
         DO 300 J = 1, NQUAL
            DENSQ = DENSQ + QUAL(J,I) * DENC(J)
 300     CONTINUE
 310     CONTINUE
         DEN(I) = DENST + DENSQ
 320  CONTINUE
 330  CONTINUE
      IF ( QJUNK ) WRITE ( KFILE, 500 ) DAY, ( I,
     *                          TEMP(I), I = 1, LSURF )
      RETURN
 500  FORMAT ( /// 5X, ‘DAY - ’, I3, 5X, ‘AFTER REFILL’ //
     *         ( 8 ( 5X, I2, 2X, F6.2 ) ) )
      END
      SUBROUTINE SETTLE
*
*  THIS SUBROUTINE UPDATES LAYER CONCENTRATIONS DUE TO QUALITY SETTLING.
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3),THEAT
      COMMON / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
      COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
      REAL LOWER
      DO 110 J = 1, NQUAL
       IF ( SETV(J) .GT. 0.0 ) THEN
      DO 100 I = 1, LSURF
*
*  DETERMINE CONTROL VOLUME.
*
      UPPER = FLOAT(I) * DELZ + SETV(J)
      LOWER = FLOAT(I-1) * DELZ + SETV(J)
      SURF  = FLOAT(LSURF-1) * DELZ + HGT(LSURF) * DELZ
*
*  CHECK FOR INTERFERENCE.
*
       IF ( UPPER .GT. SURF ) THEN
       UPPER = SURF
       END IF
*
*  UPDATE CONCENTRATIONS.
*
       IF ( LOWER .LT. SURF ) THEN
       K = I + INT(SETV(J)/DELZ)
       TOPL = FLOAT(K) * DELZ
       QUAL(J,I) = ((UPPER-TOPL)*QUAL(J,K+1)+(TOPL-LOWER)*QUAL(J,K))
     *             / DELZ
       ELSE
       QUAL(J,I) = 0.0
       END IF

  100  CONTINUE
       END IF
  110  CONTINUE

      RETURN
      END
      SUBROUTINE STABITY
*
*     COMPUTES THE STABILITY OF STRATIFICATION
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
      COMMON / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
      COMMON / CC / DEN(100),NUSURF
      COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
      PERT=HGT(LSURF)
      CP=123348.100
      CALL VOLUME(LSURF,PERT,TVOL)
*
*  COMPUTE TOTAL MASS AND TOTAL HEAT
*
      THEAT = 0.0
      TMASS=0.0
      DO 5 I=1,LSURF
      TMASS=HGT(I)*VOL(I)*DEN(I) + TMASS
      THEAT = THEAT + VOL(I)*HGT(I)*TEMP(I)*CP
    5 CONTINUE
*
*  FIND THE ELEVATION MATCHING THE
*  RESERVOIR'S AVERAGE WATER DENSITY
*
      DENA=TMASS/TVOL
      DO 10 I=1,LSURF
      IF((DEN(I)-DENA).LE.0.0)GO TO 20
   10 CONTINUE
      WRITE(6,100)DEN(I),DENA
  100 FORMAT(1X,'WARNING',10X,'DEN',F20.15,'DENA',F20.15)
   20 CONTINUE
      D=DELZ*((LSURF-1)+HGT(LSURF))
      ZC=D-(I-1)*DELZ
*
*  COMPUTE AND SUM THE MOMENTS OF DENSITY
*  VARIATION TO APPROXIMATE STABILITY
*
      STOT=0.0
      DO 30 I=1,LSURF
      ZDEL=(D-I*DELZ+DELZ/2)-ZC
      DDEL=DEN(I)-DENA
      STOT=STOT+DDEL*ZDEL*HGT(I)*VOL(I)
   30 CONTINUE
*
*  COMPUTE STABILITY
*
      SAREA=VOL(LSURF)/DELZ
      STAB=(STOT/SAREA)*929
      RETURN
      END
```

```
        SUBROUTINE STABLE
*
*  THIS SUBROUTINE MIXES WHERE AN
*  INSTABILITY EXISTS TO PRODUCE
*  A STABLE DENSITY PROFILE
*
        COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
        COMMON / BB / TEMP(100), HGT(100), VOLUME(100), TARGET, TMIX, STAB
        COMMON / CC / DEN(100), NUSURF
        COMMON / EE / KFILE, LFILE, JFILE, IFILE
        COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
        COMMON / WW / QNKWAL, YEAR(2)
        COMMON / XX / QJUNK, ONE, TWO
        DIMENSION QMIX(3), SUMQV(3)
        INTEGER DAY
        LOGICAL QJUNK, QNKWAL
        DATA C1, C2 / - 3.9863, 508929.2 /
        DATA C3, C4 / 288.9414, 68.12963 /
        DENFUN ( T ) = 1. - ( T + C1 ) ** 2 / C2
     *                 * ( T + C3 ) / ( T + C4 )
*
*  DETERMINE IF AN INSTABILITY EXISTS
*
        K = 1
 100    CONTINUE
        IF ( K .GE. LSURF ) GO TO 310
        IF ( DEN(K) .LT. DEN(K + 1) ) GO TO 110
        K = K + 1
        GO TO 100
 110    CONTINUE
*
*  SUCCESSIVELY MIX LAYERS ABOVE THE
*  LEVEL OF INSTABILITY UNTIL A STABLE
*  DENSITY GRADIENT IS OBTAINED
*
        MIXLOW = K
        MIXTOP = MIXLOW
        VOLHGT = VOLUME(MIXTOP) * HGT(MIXTOP)
        SUMTV = TEMP(MIXTOP) * VOLHGT
        IF ( QNKWAL ) GO TO 130
        DO 120 J = 1, NQUAL
        SUMQV(J) = QUAL(J, MIXTOP) * VOLHGT
 120    CONTINUE
 130    CONTINUE
        SUMVOL = VOLHGT
 140    CONTINUE
        MIXTOP = MIXTOP + 1
        VLMXTP = VOLUME(MIXTOP) * HGT(MIXTOP)
        SUMVOL = SUMVOL + VLMXTP
        SUMTV = SUMTV + TEMP(MIXTOP) * VLMXTP
        IF ( QNKWAL ) GO TO 160
        DO 150 J = 1, NQUAL
        SUMQV(J) = SUMQV(J) + QUAL(J, MIXTOP) * VLMXTP
 150    CONTINUE
 160    CONTINUE
*
*  COMPUTE TEMPERATURE AND QUALITY
*  VALUES FOR MIXED LAYERS
*
        TMIX = SUMTV / SUMVOL
        IF ( QNKWAL ) GO TO 180
        DO 170 J = 1, NQUAL
```

```
      QMIX(J) = SUMQV(J) / SUMVOL
 170  CONTINUE
 180  CONTINUE
*
*  COMPUTE DENSITY OF MIXED LAYERS
*
      DENST = DENFUN ( TMIX )
      DENSQ = 0.
      IF ( QNKWAL ) GO TO 200
      DO 190 J = 1, NQUAL
      DENSQ = DENSQ + QMIX(J) * DENC(J)
 190  CONTINUE
 200  CONTINUE
      DENMIX = DENST + DENSQ
      IF ( MIXTOP .EQ. LSURF ) GO TO 210
      IF ( DENMIX .LT. DEN(MIXTOP + 1) ) GO TO 140
 210  CONTINUE
      IF ( MIXLOW .LE. 1 ) GO TO 280
*
*  DETERMINE IF AN INSTABILITY
*  EXISTS BELOW THE MIXED LEVEL
*
      IF ( DEN(MIXLOW - 1) .GE. DENMIX ) GO TO 280
*
*  SUCCESSIVELY MIX LAYERS BELOW THE
*  MIXED LEVEL IN AN INSTABILITY EXISTS
*
      MIXLOW = MIXLOW - 1
*
*  COMPUTE TEMPERATURE AND QUALITY
*  VALUES FOR MIXED LAYERS
*
      VLMXLW = VOLUME(MIXLOW)
      SUMVOL = SUMVOL + VLMXLW
      SUMTV = SUMTV + TEMP(MIXLOW) * VLMXLW
      IF ( QNKWAL ) GO TO 230
      DO 220 J = 1, NQUAL
      SUMQV(J) = SUMQV(J) + QUAL(J, MIXLOW) * VLMXLW
 220  CONTINUE
 230  CONTINUE
      TMIX = SUMTV / SUMVOL
      IF ( QNKWAL ) GO TO 250
      DO 240 J = 1, NQUAL
      QMIX(J) = SUMQV(J) / SUMVOL
 240  CONTINUE
 250  CONTINUE
*
*  COMPUTE DENSITY OF MIXED LAYERS
*
      DENST = DENFUN ( TMIX )
      DENSQ = 0.
      IF ( QNKWAL ) GO TO 270
      DO 260 J = 1, NQUAL
      DENSQ = DENSQ + QMIX(J) * DENC(J)
 260  CONTINUE
 270  CONTINUE
      DENMIX = DENST + DENSQ
      GO TO 210
 280  CONTINUE
*
*  SET DENSITY TEMPERATURES AND
*  QUALITIES FOR MIXED LEVEL
```

```
*
        DO 300 I = MIXLOW, MIXTOP
        TEMP(I) = TMIX
        DEN(I) = DENMIX
        IF ( QNKWAL ) GO TO 300
        DO 290 J = 1, NQUAL
        QUAL(J, I) = QMIX(J)
290     CONTINUE
300     CONTINUE
        K = MIXTOP
        GO TO 100
310     CONTINUE
        IF ( QJUNK ) WRITE ( KFILE, 500 ) DAY
        IF ( QJUNK ) WRITE ( KFILE, 510 )
    *                        ( I, TEMP(I), I = 1, LSURF )
        RETURN
500     FORMAT ( /// 5X, 'DAY - ', I3, 5X, 'AFTER STABILITY' )
510     FORMAT ( /// ( 8 ( 5X, I2, 2X, F6.2 ) ) )
        END
```

```
         SUBROUTINE TPLOT
*
*  PREPARE GRAPHICAL OUTPUT
*
         COMMON / CH / NM, TITLE
         COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
    *           / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
    *           / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
    *           / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
    *           / OO / AVTEMP, AVQUAL(3), AVGT(366), AVGQ(366, 3)
    *           / PP / FIRST, LAST, NJ, INDEX(366)
    *           / VV / START, FINISH, QRELE, QPROF
    *           / WW / QNKWAL, YEAR(2)
         DIMENSION ELEV(80)
         CHARACTER TITLE*78, NM*3
         INTEGER PFILE, FIRST, YEAR, FINISH, START
         LOGICAL QFT, QFC, QFG, QNKWAL
         DATA PFILE / 10 /
         DATA ZERO / 0. /
         DATA QFT, QFC, QFG / 3 * .TRUE. /
         RETURN
*-------------------------------*
         ENTRY XGRAPH
*-------------------------------*
*
*  OUTPUT DATA
*  FOR PROFILES
*
         IF ( .NOT. QFT ) GO  TO 100
         QFT = .FALSE.
         WRITE (PFILE, 500) TITLE
 100     CONTINUE
*
*  OUTPUT TITLE
*  AND DEPTH
*
         WRITE ( PFILE, 550 )
         IF ( .NOT. QFG ) GO TO 110
         QFG = .FALSE.
         HMAX = DELZ * FLOAT ( MAXLAY )
         WRITE ( PFILE, 510 ) HMAX
 110     CONTINUE
*
*  OUTPUT DATE
*
         WRITE ( PFILE, 520 ) NJ, NM, YEAR(1)
*
*  OUTPUT NUMBER OF
*  POINTS AND QUALITIES
*
         WRITE ( PFILE, 530 ) LSURF
*
*  OUTPUT TABLE OF ELEVATION
*  TEMPERATURE AND QUALITY
*
         ELEV(1) = 0.
         ELEV(LSURF) = DEPTH
         ISM = LSURF - 1
         DO 120 I = 2, ISM
         ELEV(I) = FLOAT ( I ) * DELZ - 0.5 * DELZ
 120     CONTINUE
         DO 130 I = 1, LSURF
```

```
        IF ( NQUAL .EQ. 0 ) WRITE ( PFILE, 540 ) ELEV(I), TEMP(I)
        IF ( NQUAL .GT. 0 ) WRITE ( PFILE, 540 ) ELEV(I), TEMP(I),
     *                          ( QUAL(J, I), J = 1, NQUAL )
 130    CONTINUE
        RETURN
*-------------------------------*
        ENTRY XCYCLE
*-------------------------------*
*
*  OUTPUT DATA TO PLOT
*  RELEASE TEMPERATURES
*
        IF ( .NOT. QFT ) GO TO 140
        QFT = .FALSE.
        WRITE ( PFILE, 500 ) TITLE
 140    CONTINUE
        WRITE ( PFILE, 560 )
        IF ( .NOT. QFC ) GO TO 150
        QFC = .FALSE.
        WRITE ( PFILE, 580 ) START, FINISH
 150    CONTINUE
        WRITE ( PFILE, 570 ) YEAR
        WRITE ( PFILE, 540 ) ( AVGT(N), N = START, FINISH )
        IF ( QNKWAL ) GO TO 170
        DO 160 J = 1, NQUAL
        WRITE ( PFILE, 590 ) J
        WRITE ( PFILE, 540 ) ( AVGQ(N, J), N = START, FINISH )
 160    CONTINUE
 170    CONTINUE
        RETURN
 500    FORMAT ( A )
 510    FORMAT ( 'DEPTH', 5X, 2F10.0 )
 520    FORMAT ( 'DATE', 6X, I2, 1X, A3, 1X, I4 )
 530    FORMAT ( 'POINTS', 4X, I10, I5 )
 540    FORMAT ( 8F10.3 )
 550    FORMAT ( 'PROFILES' )
 560    FORMAT ( 'RELEASE VALUES' )
 570    FORMAT ( 'YEAR', 6X, I5 )
 580    FORMAT ( 'SIMULATION', 2I5 )
 590    FORMAT ( 'QUALITY', 3X, I5 )
        END
```

```
      SUBROUTINE VOLUME(ITOP,PER,TVOL)
*
*  DETERMINE STORAGE VOLUME FOR A GIVEN TOP LAYER AND PROPORTION FILLED.
*
       COMMON / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
       TVOL = 0.0
       ITM1 = ITOP -1
       DO 10 I=1,ITM1
       TVOL = TVOL + VOL(I)
   10  CONTINUE
       TVOL = TVOL + PER * VOL(ITOP)
       RETURN
       END
```

```
         SUBROUTINE VPORT
*
*  CALCULATE WITHDRAWAL LIMITS
*  AND VELOCITY PROFILE FOR ORIFICE
*
         COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
         COMMON / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
         COMMON / CC / DEN(100), NUSURF
     *          / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
     *          / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
     *          / DE / TOPLIM, LOWLIM, V(100),FLORAT,LL(8),LT(8)
     *          / EF / AREA, HGTPRT, LAYPRT, DENPRT
     *          / XX / QJUNK, ONE, TWO
     *          / IJ / WANGLE, WTHETA(8)
         COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
*
         LOGICAL  QBLIM, QTLIM, QMETR, QSINK1, QSINK2, QSHIFT
*
         INTEGER XXX , TOPLIM
*
         CHARACTER*4 XDUMY, XDUMY1, XDUMY2, XDUMY3
         CHARACTER*6 SUBR
*
         DATA  MAX, VMAX, TINY                / 10, 1., 1.0E-08 /
         DATA XDUMY, XDUMY1, XDUMY2, XDUMY3 / 4 * '0    '       /
         DATA SUBR                            / 'VPORT'         /
         DATA G                               / 32.18           /
*
*.... FUNCTION STATEMENTS TO SOLVE FOR ORIFICE WITHDRAWAL LIMITS
*     FOR INTERMEDIATE FLOW CONDITIONS ....
*
         LAYER    ( X ) = 1. + X / DELZ
         ZEE      ( X ) = ABS ( HGTPRT - X )
         FROUD    ( X ) = SQRT ( G * ABS ( 1. - DENINT ( X )
     *                    / DENPRT ) )
*
*.... COMPUTE THE DIFFERENCE BETWEEN THE FLOW 'ENERGY' AND THE
*     ENERGY EXPENDED BASED ON BOHAN AND GRACE (1969), MODIFIED
*     TO INCLUDE WITHDRAWAL ANGLE CONCEPTS ....
*
         QBNG     ( X ) = FLOCFS / PHIFRAC - C2 * FROUD ( X )
     *                    * ZEE ( X ) ** 2.5
*
*.... FUNCTION STATEMENTS TO SOLVE FOR THE UNBOUNDED WITHDRAWAL
*     LIMIT WHEN THERE IS BOTTOM OR SURFACE INTERFERENCE ....
*
          DPRIME  ( X ) = ABS ( BONLIM - X )
          BDRATIO ( X ) = SMALLB / DPRIME ( X ) /
     *                    ( 1 - SMALLB / DPRIME ( X ) )
          FROUDE  ( X ) = SQRT ( G * ABS ( 1. - DENINT ( X )
     *                   / DENPRT ) / ZEE ( X ) )
          CHI     ( X ) = 1. / 2. * ( 1. + BDRATIO ( X ) )
          PHI     ( X ) = 1. / 2. * ( 1 + 1 / PI
     *                    * SIN ( BDRATIO ( X ) * PI ) +
     *                    BDRATIO ( X ) )
*
*.... COMPUTE THE DIFFERENCE BETWEEN THE FLOW 'ENERGY' AND THE
*     'ENERGY' EXPENDED BASED ON SMITH, ET AL (1985), EQN 36 ....
*
          QSMITH  ( X ) = FLOCFS - C2 *  FROUDE ( X ) * PHI ( X )
     *                    / ( 2.0 * CHI ( X ) ) ** 3
     *                    * DPRIME ( X ) ** 3
```

```
*
*.... TOLERANCE, 10% OF LAYER THICKNESS ....
*
      SMALL = .10 * DELZ
*
*....  INITIALIZE LOGICAL VARIABLES ....
*
      QSINK1 = .TRUE.
      QSINK2 = .TRUE.
      QSHIFT = .FALSE.
*
*....  SET THE VALUE  OF THE ANGLE OF WITHDRAWAL COEFFICIENT
*      FOR THE BOUNDARY INTERFERENCE EQUATION ....
*
*.... CHECK TO SEE IF ENTERING FROM SUBROUTINE SHIFT ....
*
      IF ( QSHIFT ) GO TO 185
      PI      = 3.14159
      C2      = WANGLE / PI
      PHIFRAC = 1.0
*
*.... CONVERT TO CFS FOR CALCULATIONS....
*
      FLOCFS = FLORAT / 1.9835E-3
*
*.... CHECK FOR BOUNDARY INTERFERENCE FROM SURFACE OR BOTTOM
*     USING INTERMEDIATE FLOW EQUATION ....
*
      DENPRT = DENINT ( HGTPRT )
      DENBOT = DENINT ( 0.     )
      DENUPP = DENINT ( DEPTH  )
      IF ( HGTPRT .GT. 0.0 ) THEN
          QBLIM  = QBNG   ( 0.    ) .GE. 0.
      ELSE
          QBLIM  = .TRUE.
      ENDIF
      QTLIM  = QBNG   ( DEPTH ) .GE. 0.
*
*.... DIRECT COMPUTATIONS BASED ON INTERFERENCE
*     CHARACTERISTICS ....
*
      IF ( QTLIM .AND. QBLIM            ) GO TO 540
      IF ( QTLIM                        ) GO TO 500
      IF ( QBLIM                        ) GO TO 510
      IF ( .NOT. QTLIM .AND. .NOT. QBLIM ) GO TO 540
 500  CONTINUE
*
*....  IF ONLY ONE  BOUNDARY EXPERIENCES INTERFERENCE, FIND THE
*      HEIGHT OF WITHDRAWAL USING SMITH, 1987 ....
*
*.... DETERMINE THE HEIGHT OF THE TRUNCATED PORTION, THE BOUNDARY
*     LIMIT, THE SEARCH INTERVAL LIMITS, AND THE FUNCTION SIGN AT
*     THE SEARCH LIMITS ....
*
*.... SURFACE INTERFERENCE ....
*
      SMALLB = DEPTH - HGTPRT
      DENLIM = DENUPP
      BONLIM = DEPTH
      X1     = 0.
      X2     = DEPTH
      H1     = QSMITH ( X1 )
```

```
         GO TO 530
 510     CONTINUE
*
*.... BOTTOM INTERFERENCE ....
*
         TRUNCZ = HGTPRT
         DENLIM = DENBOT
         BONLIM = 0.
         X1     = 0.
         X2     = DEPTH
         H1     = 1.
 530     CONTINUE
*
*.... FIND THE LIMIT USING A HALF-INTERVAL SEARCH ....
*
*.... INITIALIZE X3 ....
*
         X3 = SMALL
*
*.... BEGIN ITERATION ....
*
         DO 560 I = 1, 2 * MAX
            X4 = X3
*
*.... ESTABLISH A THIRD POINT BETWEEN TWO EXISTING POINTS ....
*
            X3 = ( X1 + X2 ) / 2.0
*
*.... CALCULATE FUNCTION SIGN AT NEW POINT ....
*
            H3 = QSMITH ( X3 )
            ZONED = ABS ( BONLIM - X3 )
*
*.... IF NEW POINT IS SAME AS PREVIOUS POINT (WITHIN TOLERANCE),
*     ITERATION IS COMPLETE ....
*
            IF ( ABS ( X4 - X3 ) .LT. SMALL ) GO TO 570
*
*.... USE AS NEW SEARCH LIMITS THE MOST RECENT POINT AND THE
*     REMAINING POINT OF OPPOSITE FUNCTION SIGN ....
*
            IF ( H1 * H3  .LT. 0. ) GO TO 535
            X1 = X3
            H1 = H3
            GO TO 560
 535        CONTINUE
            X2 = X3
 560     CONTINUE
*
*.... CONVERGENCE WAS NOT REACHED ....
*
         CALL ERROR ( 1500 )
 570     CONTINUE
         PHIFRAC = PHI ( X3 )
*
*....  CALCULATE WITHDRAWAL LIMIT ....
*
         IF ( QTLIM .AND. .NOT. QBLIM ) HGTLOW = DEPTH - ZONED
         IF ( QBLIM .AND. .NOT. QTLIM ) HGTTOP = ZONED
 540     CONTINUE
*
*.... USAGE FOR THE BOHAN AND GRACE EQUATION
```

```
*     1. NO BOUNDARY INTERFERENCE
*     2. BOTH BOUNDARIES INTERFERE WITH WITHDRAWAL ZONE
*     3. SINGLE BOUNDARY INTERFERENCE.  THEORETICAL LIMIT
*        OF ONE INTERFERED WITH MUST BE DETERMINED
*        (FREE LIMIT IS DETERMINED ABOVE WITH SMITH EQUATION) ....
*
         IF ( QTLIM .AND. .NOT. QBLIM  ) GO TO 150
*
*.... EMBARK ON DETERMINATION OF LOWER WITHDRAWAL LIMIT ....
*
*.... IF LOWER LIMIT IS WITHIN THE POOL THEN FIND IT WITH A
*     HALF-INTERVAL SEARCH ....
*
*.... INITIAL SEARCH LIMITS ARE X1 =0 AND X2 = HGTPRT ....
*
         X1 = 0.0
*
*.... IF BOTTOM BOUNDARY INTERFERENCE EXISTS (LOWER LIMIT OUTSIDE
*     POOL), THEN X1 = - DEPTH ....
*
         IF ( QBLIM ) X1 = - DEPTH
         F1 = QBNG ( X1 )
         DENLIM = DENBOT
         X2 = HGTPRT
         X3 = -2. * SMALL
         ASSIGN 140 TO XXX
 110     CONTINUE
*
*.... INITIATE ITERATION PROCESS ....
*
         DO 130 I = 1, MAX
*
*.... ESTABLISH A THIRD POINT BETWEEN THE TWO EXISTING POINTS ....

            X4 = X3
            X3 = ( X1 + X2 ) / 2.
*
*.... CALCULATE FUNCTION SIGN AT NEW ELEVATION ....
*
            DENLIM = DENINT ( X3 )
            IF (  DENLIM .EQ. DENPRT ) GO TO XXX
            F3 = QBNG ( X3 )
*
*....  IF NEW POINT IS SAME AS PREVIOUS POINT (WITHIN TOLERANCE)
*      THEN SEARCH IS COMPLETE ....
*
            IF ( ABS ( X4 - X3 ) .LT. SMALL )
     *           GO TO XXX, ( 140, 170 )
*
*.... USE AS NEW SEARCH LIMITS THE MOST RECENTLY COMPUTED POINT AND
*     THE REMAINING POINT OF OPPOSITE SIGN
*
            IF ( F1 * F3 .GT. 0. ) GO TO 120
            X2 = X3
            GO TO 130
 120        CONTINUE
            X1 = X3
            F1 = F3
 130     CONTINUE
*
*.... CONVERGENCE HAS NOT BEEN REACHED ....
```

```
        CALL ERROR ( 1510 )
 140    CONTINUE
*
*.... SET LOWER LIMIT ELEVATION ....
*
        HGTLOW = X3
 150    CONTINUE
        IF ( QBLIM .AND. .NOT. QTLIM ) GO TO 180
*
*.... APPLY SAME PROCEDURE FOR DETERMINING UPPER WITHDRAWAL LIMIT
*     FOR ORIFICE
*
*.... DETERMINE ELEVATION, LAYER AND FUNCTION SIGN AT SEARCH
*     LIMITS.  IF NEITHER LIMIT EXPERIENCES INTERFERENCE THE
*     INITIAL SEARCH LIMITS ARE X1 = HGTPRT AND X2 = DEPTH.
*     HOWEVER, IF SURFACE INTERFERENCE EXISTS ( UPPER LIMIT OUTSIDE
*     POOL ), THEN X2 = 2. * DEPTH ....
*
        X1 = HGTPRT
        X2 = DEPTH
        IF ( QTLIM )  X2 = 2 * DEPTH
        F1 = QBNG ( X1 )
*
*.... USE THE PRIOR SEARCH PROCEDURE ....
*
        ASSIGN 170 TO XXX
        DENLIM = DENUPP
        GO TO 110
 170    CONTINUE
        HGTTOP = X3
 180    CONTINUE
*
*.... CALCULATE LOCATION OF MAXIMUM VELOCITY AND THICKNESS OF
*     WITHDRAWAL LIMITS ....
*
 185    CONTINUE
        ZONE = HGTTOP - HGTLOW
        ZTOP = HGTTOP - HGTPRT
        ZLOW = HGTPRT - HGTLOW
*
*.... BASED ON BOHAN AND GRACE
*
        YVMAX = ZONE * ( SIN ( 1.57 * ZLOW / ZONE ) ) ** 2
*
*.... HEIGHT ABOVE BOTTOM.  DO NOT PERMIT MAX VELOCITY
*     OUTSIDE THE POOL ....
*
        XVMAX = YVMAX + HGTLOW
        IF ( XVMAX .LT. 0.0   ) XVMAX = 0.0
        IF ( XVMAX .GT. DEPTH ) XVMAX = DEPTH
        LVMAX = LAYER ( XVMAX )
*
*.... MAXIMUM VELOCITY OUTSIDE THE POOL ....
*
        IF ( ( XVMAX .LT. 0. ) .OR. ( XVMAX .GT. DEPTH ) )
     *        CALL ERROR ( 1520 )
*
*.... ASSIGN DENSITIES AT LIMITS AND MAXIMUM VELOCITY ....
*
        DVMAX  = DENINT ( XVMAX  )
        DENLOW = DENINT ( HGTLOW )
        DENTOP = DENINT ( HGTTOP )
```

```
*
*.... WITHDRAWAL LAYER LIMITS ....
*
         IF ( HGTLOW .LT. 0.    ) LOWLIM = LAYER ( 0.     )
         IF ( HGTLOW .GE. 0.    ) LOWLIM = LAYER ( HGTLOW )
         IF ( HGTTOP .GE. DEPTH ) TOPLIM = LSURF
         IF ( HGTTOP .LT. DEPTH ) TOPLIM = LAYER ( HGTTOP )
*
*.... ZERO THE VELOCITY PROFILE FOR THE CURRENT PORT ....
*
         DO 190 I = 1, LSURF
            V ( I ) = 0.
 190     CONTINUE
*
*.... IF LOWER WITHDRAWAL LAYERS ARE OF CONSTANT DENSITY THEN
*     ASSIGN CONSTANT VELOCITY TO EACH LAYER ....
*
         DENDIF = DENLOW - DVMAX
         IF ( DENDIF .GT. 0. ) GO TO 210
         DO 200 I = LOWLIM, LVMAX
            V ( I ) = VMAX
 200     CONTINUE
         GO TO 240
 210     CONTINUE
*
*.... CALCULATE VELOCITY PROFILE FROM LAYER OF MAXIMUM VELOCITY
*     TO LOWER LIMIT ....
*
         IF ( LOWLIM .EQ. LVMAX ) GO TO 240
         DO 230 I = LOWLIM, LVMAX
            Y1 = DELZ * ( LVMAX - I )
            DELDEN = DEN ( I ) - DVMAX
*
*.... BASED ON BOHAN AND GRACE ....
*
            RATIO = Y1 * DELDEN / ( ZLOW * DENDIF )
            RATIO = AMIN1 ( 1., RATIO )
            V(I) = VMAX * ( 1. - RATIO ) ** 2.0
 230     CONTINUE
 240     CONTINUE
*
*.... IF UPPER WITHDRAWAL LAYERS ARE OF CONSTANT DENSITY THEN
*     ASSIGN CONSTANT VELOCITY TO EACH LAYER ....
*
         DENDIF = DVMAX - DENTOP
         IF ( DENDIF .GT. 0. ) GO TO 260
         DO 250 I = LVMAX, TOPLIM
            V ( I ) = VMAX
 250     CONTINUE
         GO TO 290
 260     CONTINUE
*
*.... DETERMINE VELOCITY PROFILE FROM LAYER OF MAXIMUM VELOCITY
*     TO UPPER LIMIT ....
*
         IF ( LVMAX .EQ. TOPLIM ) GO TO 290
         DO 280 I = LVMAX, TOPLIM
            Y1 = DELZ * ( I - LVMAX )
            DELDEN = DVMAX - DEN ( I )
*
*.... BASED ON BOHAN AND GRACE ....
*
```

```
            RATIO = Y1 * DELDEN / ( ZTOP * DENDIF )
            RATIO = AMIN1 ( 1., RATIO )
            V(I) = VMAX * ( 1. - RATIO ) ** 2.0
 280     CONTINUE
 290     CONTINUE
*
*.... CHECK FOR POINT SINK DESCRIPTION.  ASSUME SQUARE PORTS. ....
*
         VDIM = SQRT ( AREA )
         VDIM2 = VDIM / 2.
         PRTTOP = HGTPRT + VDIM2
         VD2 = VDIM2
         IF ( PRTTOP .GT. DEPTH ) VD2 = DEPTH - HGTPRT
         IF ( PRTTOP .GT. DEPTH ) PRTTOP = DEPTH
         PRTBOT = HGTPRT - VDIM2
         IF ( PRTBOT .LT. 0. .AND. PRTBOT .GT. -.1 ) PRTBOT = 0.
         DRPTOP = DENPRT - DENINT ( PRTTOP )
         DRPBOT = DENINT ( PRTBOT ) - DENPRT
*
         DRTLIM = DENPRT - DENTOP
         DRBLIM = DENLOW - DENPRT
*
         IF ( DRPBOT .LT. TINY ) DRPBOT = TINY
         IF ( DRPTOP .LT. TINY ) DRPTOP = TINY
         IF ( DRBLIM .LT. TINY ) DRBLIM = TINY
         IF ( DRTLIM .LT. TINY ) DRTLIM = TINY
         IF ( VDIM2  .LT. TINY ) VDIM2  = TINY
         IF ( VD2    .LT. TINY ) VD2    = TINY
*
*.... EMPIRICAL EQUATIONS FOR POINT SINK VERIFICATION ....
*
         SINK1  = ( DRBLIM ) * ZLOW / ( DRPBOT * VDIM2 )
         SINK2  = ( DRTLIM ) * ZTOP / ( DRPTOP * VD2 )
         QSINK1 = SINK1 .GT. 3.0
         QSINK2 = SINK2 .GT. 3.0
         RETURN
         END
```

```
        SUBROUTINE VWEIR
*
*  CALCULATE WITHDRAWAL LIMITS AND
*  VELOCITY PROFILE FOR WEIR FLOW
*
        COMMON / CC / DEN(100), NUSURF
     *         / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
     *         / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
     *         / XX / QJUNK, ONE, TWO
     *         / BC / WRLNG, WRHGT, WRTYPE, DCOEF
     *         / DE / TOPLIM, LOWLIM, V(100), FLORAT, LL(8), LT(8)
        LOGICAL QFIRST, QJUNK, QBLIM, QTLIM, Q1, Q2
        INTEGER TOPLIM
        CHARACTER*4 XFREE, XSUBM, WRTYPE
        DATA QFIRST, A, B / .TRUE., 4.35, - 1.04 /
        DATA G, VMAX, ITMAX / 32.18, 1., 10 /
        DATA XFREE, XSUBM / 'FREE', 'SUBM' /
        DATA SMALL / 1. /
*
*  FUNCTION STATEMENTS TO SOLVE
*  FOR LOWER WITHDRAWALS LIMIT
*
        LAYER ( X ) = 1. + X / DELZ
        SIZE ( X ) = ABS ( WRHGT - X )
        RWEIR ( Z ) = SQRT ( Z ) * ( 1. + Z / HEAD )
        FWEIR ( X ) = AVGVEL - .32 * RWEIR ( SIZE ( X ) ) *
     *                SQRT ( G * ABS ( 1. - DENINT ( X ) / WRDEN ) )
        IF ( .NOT. QFIRST) GO TO 100
        QFIRST = .FALSE.
        EXPNT = A + B * DCOEF
        IF ( ABS ( DCOEF - 3.00 ) .LT. .01 ) EXPNT = 1.5
        IF ( ABS ( DCOEF - 3.33 ) .LT. .01 ) EXPNT = 0.5
        IF ( ABS ( DCOEF - 4.10 ) .LT. .01 ) EXPNT = 0.2
 100    CONTINUE
*
*  CALCULATE AVERAGE VELOCITY
*  OVER THE WEIR IN FT/SEC
*
        VMAX = 1.
        HEAD = DEPTH - WRHGT
        FLOCFS = FLORAT / 1.9835E-3
        AVGVEL = FLOCFS / ( HEAD * WRLNG )
*
*  CHECK FOR INTERFERENCE FROM
*  BOTTOM. SURFACE WILL EXHIBIT
*  INTERFERENCE.
*
        WRDEN = DENINT ( CREST )
        QBLIM = FWEIR ( 0. ) .GE. 0.
        QTLIM = .TRUE.
*
*  EMBARK ON DETERMINATION OF
*  LOWER WITHDRAWAL LIMIT
*
        IF ( .NOT. QBLIM ) GO TO 110
*
*  IF BOTTOM INTERFERENCE EXISTS THEN
*  SET LOWER LIMIT AT THE BOTTOM
*
        HGTLOW = 0.
        LOWLIM = 1
        GO TO 150
 110     CONTINUE
*
*  IF LOWER LIMIT IS WITHIN THE POOL THEN
*  FIND IT WITH A HALF - INTERVAL SEARCH
*
*  DETERMINE ELEVATION, LAYER, FUNCTION VALUE,
*  AND FUNCTION SIGN AT EACH SEARCH LIMIT
*  (1) BOTTOM OF POOL AND (2) WEIR ELEVATION
*
         X1 = 0.
         F1 = FWEIR ( X1 )
         Q1 = F1 .GT. 0.
         F2 = AVGVEL
         Q2 = F2 .GE. 0.
         X3 = - 2. * SMALL
*
*  FUNCTION MUST BE POSITIVE AT THE WEIR
*  LEVEL AND NEGATIVE AT THE BOTTOM
*
         IF ( Q1 .OR. .NOT. Q2 ) CALL ERROR ( 2060 )
*
*  INITIATE ITERATION PROCESS
*
         DO 130 I = 1, ITMAX
*
*  ESTABLISH A THIRD POINT BETWEEN
*  THE TWO EXISTING POINTS
*
         X4 = X3
         X3 = ( X1 + X2 ) / 2.
*
*  CALCULATE FUNCTION SIGN AT NEW ELEVATION
*
         F3 = FWEIR ( X3 )
*
*  IF NEW POINT IS THE SAME AS A PREVIOUS
*  POINT THEN SEARCH IS COMPLETE
*
         IF ( ABS ( X4 - X3 ) .LT. SMALL ) GO TO 140
*
*  USE AS NEW SEARCH LIMITS THE MOST
*  RECENTLY COMPUTED POINT AND THE
*  REMAINING POINT OF OPPOSITE SIGN
*
         IF ( F1 * F3 .GT. 0. ) GO TO 120
         X2 = X3
         F2 = F3
         GO TO 130
 120     CONTINUE
         X1 = X3
         F1 = F3
 130     CONTINUE
*
*  CONVERGENCE HAS NOT BEEN REACHED
*  I WONDER WHAT TO DO NOW
*
         CALL ERROR ( 2070 )
 140     CONTINUE
*
*  SET LOWER LIMIT ELEVATION AND LAYER
*
         HGTLOW = X3
```

```
      LOWLIM = LAYER ( X3 )
 150  CONTINUE
*
*  SET UPPER LIMIT AT SURFACE
*
      HGTTOP = DEPTH
      TOPLIM = NUSURF
*
*  CALCULATE LOCATION OF MAXIMUM VELOCITY
*
      ZONE = HGTTOP - HGTLOW
      ZLOW = CREST  - HGTLOW
      IF ( WRTYPE .EQ. XFREE ) YVMAX = HGTTOP
      IF ( WRTYPE .EQ. XSUBM ) YVMAX = ZONE *
     *      SIN ( 1.57 * ZLOW / ZONE )
*
*  COMPUTE THICKNESS OF WITHDRAWAL ZONE
*
      XVMAX = YVMAX + HGTLOW
      LVMAX = LAYER ( XVMAX )
      DVMAX = DENINT ( XVMAX )
      YLOW = DELZ * FLOAT ( LVMAX - LOWLIM )
      YTOP = DELZ * ( TOPLIM - FLOAT ( LVMAX ) )
      DENLOW = DENINT ( HGTLOW )
      DENTOP = DENINT ( HGTTOP )
*
*  ZERO THE VELOCITY PROFILE
*  FOR THE CURRENT WEIR
*
      DO 160 I = 1, NUSURF
      V(I) = 0.
 160  CONTINUE
*
*  IF LOWER WITHDRAWAL LAYERS ARE OF
*  CONSTANT DENSITY THEN ASSIGN
*  CONSTANT VELOCITY TO EACH LAYER
*
      DENDIF = DENLOW - DVMAX
      IF ( DENDIF .GT. 0. ) GO TO 180
      DO 170 I = LOWLIM, LVMAX
      V(I) = VMAX
 170  CONTINUE
      GO TO 210
 180  CONTINUE
*
*  CALCULATE VELOCITY PROFILE FROM LAYER
*  OF MAXIMUM VELOCITY TO LOWER LIMIT
*
      DO 200 I = LOWLIM, LVMAX
      Y = DELZ * FLOAT ( LVMAX - I )
      DELDEN = DEN(I) - DVMAX
      RATIO = Y * DELDEN / ( YLOW * DENDIF )
      IF ( QBLIM ) GO TO 190
      P = 3.0
      IF ( WRTYPE .EQ. XSUBM ) V(I) = VMAX * ( 1. - RATIO ) ** P
      IF ( WRTYPE .EQ. XFREE ) V(I) = VMAX * ( 1. - RATIO ** EXPNT )
      GO TO 200
 190  CONTINUE
      V(I) = VMAX * ( 1. - RATIO ** 2 )
 200  CONTINUE
      V(I) = 0.5 * V(2)
 210  CONTINUE
```

```
*
*  IF UPPER WITHDRAWAL LAYERS ARE
*  OF CONSTANT DENSITY THEN ASSIGN
*  CONSTANT VELOCITY TO EACH LAYER
*
       IF ( WRTYPE .EQ. XFREE ) GO TO 260
       DENDIF = DVMAX - DENTOP
       IF ( DENDIF .GT. 0. ) GO TO 230
       DO 220 I = LVMAX, TOPLIM
       V(I) = VMAX
 220   CONTINUE
       GO TO 260
 230   CONTINUE
*
*  DETERMINE VELOCITY PROFILE FROM LAYER
*  OF MAXIMUM VELOCITY TO UPPER LIMIT
*
       DO 250 I = LVMAX, TOPLIM
       Y = DELZ * FLOAT ( I - LVMAX )
       DELDEN = DVMAX - DEN(I)
       RATIO = Y * DELDEN / ( YTOP * DENDIF )
       IF ( QTLIM ) GO TO 240
       V(I) = VMAX * ( 1. - RATIO ) ** 2
       GO TO 250
 240   CONTINUE
       V(I) = VMAX * ( 1. - RATIO ** 2 )
 250   CONTINUE
       V(NUSURF) = 0.5 * V(NUSURF - 1)
 260   CONTINUE
       RETURN
       END
```

```
        SUBROUTINE XFIRST
*
*  PRINT SYSTEM CONTROL PARAMETERS
*
        COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
        COMMON / CH / NM, TITLE
        COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
        COMMON / EE / KFILE, LFILE, JFILE, IFILE
        COMMON / FF / NPORTS, PAREA(8), PHGT(8)
        COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP,  WTHDRW(100)
        COMMON / II / EK, ET, SHORT, BETA, LAMBDA, MIXCOEF, GAMMA,
    *                  DECAY
        COMMON / KK / NWELL(8), FMAX(8), FMIN(8), SELMAX
        COMMON / LL / FGAREA, FGHGT, FGMAX, FGMIN, FGANG, LPORT(8)
        COMMON / PP / FIRST, LAST, NJ, INDEX(366)
        COMMON / QQ / QVERI, QOUTC, QINTC, QINCFS, QOCFS, QWCFS
        COMMON / VV / START, FINISH, QRELE, QPROF
        COMMON / WW / QNKWAL, YEAR(2)
        CHARACTER TITLE*78, NM*3
        DIMENSION XMAX(8), XMIN(8)
        LOGICAL QVERI, QNKWAL
        INTEGER START, FINISH, FIRST, YEAR
        REAL LAMBDA, MIXCOEF
        DATA FACTOR / 1.9835E - 03 /
        WRITE ( LFILE, 500 )
        WRITE ( LFILE, 510 ) TITLE
        IF ( YEAR(1) .GT. 0 ) WRITE ( LFILE, 520 ) YEAR
        IF ( QVERI ) WRITE ( LFILE, 540 )
        IF ( .NOT. QVERI ) WRITE ( LFILE, 530 )
        IF ( .NOT. QNKWAL ) WRITE ( LFILE, 550 ) NQUAL
        WRITE ( LFILE, 560 ) NIP
        WRITE ( LFILE, 570 ) MAXLAY
        WRITE ( LFILE, 580 ) DELZ
        WRITE ( LFILE, 590 ) BOTTOM
        WRITE ( LFILE, 600 ) NPORTS
        WRITE ( LFILE, 620 ) BETA
        WRITE ( LFILE, 630 ) LAMBDA
        WRITE ( LFILE, 632 ) MIXCOEF
        WRITE ( LFILE, 634 ) DECAY
        WRITE ( LFILE, 636 ) GAMMA
        WRITE ( LFILE, 640 ) FIRST, LAST
        WRITE ( LFILE, 650 ) START, FINISH
*
*  COMPUTE LOCATION OF PORTS
*
        DO 100 K = 1, NPORTS
        LPORT(K) = 1. + PHGT(K) / DELZ
 100    CONTINUE
        IF ( QVERI ) GO TO 120
*
*  CONVERT CAPACITIES TO
*  CFS UNITS FOR PRINTOUT
*
        DO 110 K = 1, NPORTS
        XMAX(K) = FMAX(K) / FACTOR
        XMIN(K) = FMIN(K) / FACTOR
 110    CONTINUE
        LPORT(NPORTS + 1) = 1. + FGHGT / DELZ
        FXMAX = FGMAX / FACTOR
        FXMIN = FGMIN / FACTOR
        SXMAX = SELMAX / FACTOR
        WRITE ( LFILE, 660 )
        WRITE ( LFILE, 670 )
        WRITE ( LFILE, 680 ) ( K, PAREA(K), PHGT(K), XMAX(K),
    *           XMIN(K), NWELL(K), LPORT(K), K = 1, NPORTS )
        WRITE ( LFILE, 690 ) FGAREA, FGHGT, FXMAX,
    *                        FXMIN, LPORT(NPORTS + 1)
        WRITE ( LFILE, 700 ) SXMAX
120     CONTINUE
        WRITE ( LFILE, 710 ) DEPTH
        IF ( .NOT. QVERI ) RETURN
        WRITE ( LFILE, 720 )
        WRITE ( LFILE, 730 )
        WRITE ( LFILE, 740 ) ( K, PAREA(K), PHGT(K),
    *                          LPORT(K), K = 1, NPORTS )
        RETURN
500     FORMAT ( '1' )
510     FORMAT ( /// 15X, A )
520     FORMAT ( / 15X, I4, ' - HYDROLOGY' //
    *            15X, I4, ' - METEOROLOGY' )
530     FORMAT ( / 15X, 'PREDICTION MODE' )
540     FORMAT ( / 15X, 'VERIFICATION MODE' )
550     FORMAT ( / 15X,  'QUALITIES', 20X, I1 )
560     FORMAT ( / 15X, 'INFLOW POINTS', 16X, I1 )
570     FORMAT ( / 15X, 'MAXIMUM LAYERS', 13X, I3 )
580     FORMAT ( / 15X, 'LAYER THICKNESS', 12X,
    *              F3.0, 5X, 'FEET' )
590     FORMAT ( / 15X, 'BOTTOM ELEVATION', 9X,
    *              F5.0, 5X, 'FEET' )
600     FORMAT ( / 15X, 'NUMBER OF PORTS', 14X, I1 )
620     FORMAT ( / 15X, 'BETA', 22X, F4.2 )
630     FORMAT ( / 15X, 'LAMBDA', 20X, F4.2 )
632     FORMAT ( / 15X, 'MIXING COEF.', 13X, F5.2 )
634     FORMAT ( / 15X, 'MIXING DECAY COEF.', 8X, E8.1 )
636     FORMAT ( / 15X, 'ENTRAINMENT  COEF.', 8X, F4.2 )
640     FORMAT ( / 15X, 'DATA INTERVAL', 14X, I3, '  -  ', I3 )
650     FORMAT ( / 15X, 'SIMULATION INTERVAL',
    *               8X, I3, '  -  ', I3 )
660     FORMAT ( /// 16X, 'PORT', 4X, 'PORT AREA',
    *               6X, 'PORT HEIGHT', 6X, 'MAXIMUM FLOW',
    *               3X, 'MINIMUM FLOW', 3X, 'WETWELL',
    *               5X, 'LAYER' )
670     FORMAT (   15X, 'NUMBER', 3X, '(SQ. FT.)',
    *              3X, '(FT. FROM BOTTOM)', 7X, '(CFS)',
    *             10X, '(CFS)', 7X, 'NUMBER', 5X, 'NUMBER' )
680     FORMAT ( / ( 17X, I2, 5X, F6.0, 11X, F5.1,
    *                10X, F8.0, 9X, F6.0, 10X, I1, 8X, I3 ) )
690     FORMAT ( 16X, 'FLOOD' / 17X, 'GATE', 2X, F7.1,
    *            11X, F5.1, 9X, F9.1, 9X, F6.1, 19X, I3 )
700     FORMAT ( //// 15X, 'SELECTIVE WITHDRAWAL CAPACITY',
    *             5X, F10.0, 5X, 'CFS' )
710     FORMAT ( / 15X, 'INITIAL DEPTH', 11X, F6.2, 5X, 'FEET' )
720     FORMAT ( /// 16X, 'PORT', 4X, 'PORT AREA',
    *                 6X, 'PORT HEIGHT', 9X, 'LAYER' )
730     FORMAT (     15X, 'NUMBER', 3X, '(SQ. FT.)',
    *                 3X, '(FT. FROM BOTTOM)', 5X, 'NUMBER' )
740     FORMAT ( / ( 17X, I2, 5X, F6.0, 11X, F5.1, 14X, I2 ) )
        END
```

```
      SUBROUTINE XPRINT
*
*  PRINT SUMMARY INFORMATION
*  FOR EVERY DAY OF SIMULATION
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
      COMMON / BB / TEMP(100), HGT(100), VOL(100), TARGET, TMIX, STAB
      COMMON / CH / NM, TITLE
      COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
      COMMON / DE / TOPLIM, LOWLIM, V(100), FLORAT, LL(8), LT(8)
      COMMON / EE / KFILE, LFILE, JFILE, IFILE
      COMMON / FF / NPORTS, PAREA(8), PHGT(8)
      COMMON / GG / FLOWIN(3), TEMPIN(3), QUALIN(3, 3)
      COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
      COMMON / JJ / SUMOUT, WIDTH(100)
      COMMON / MM / DAY, PORT(8), PHLOW(8), OPEN
      COMMON / OO / AVTEMP, AVQUAL(3), AVGT(366), AVGQ(366, 3)
      COMMON / PP / FIRST, LAST, NJ, INDEX(366)
      COMMON / QQ / QVERI, QOUTC, QINTC, QINCFS, QOCFS
      COMMON / UU / ENFLOW(100), SUMFLO(366), DPRINT(14)
      COMMON / VV / START, FINISH, QRELE, QPROF
      COMMON / WW / QNKWAL, YEAR(2)
      COMMON / XX / QJUNK, ONE, TWO
      LOGICAL QVERI, QNKWAL, QHEAD, QPDAY, QRELE, QPROF, QFIRST
      LOGICAL QJUNK
      INTEGER START, FINISH, PORT, OPEN, YEAR,
     *        FIRST, DPRINT, DAY, STORE, SAVE
      CHARACTER TITLE*78, NM*3
      DIMENSION STORE(0:14)
      DATA MAXLIN / 40 /
      DATA QFIRST / .TRUE. /
      DATA NPDAYS / 14 /
*
*  INSURE THAT PRINT
*  DAYS ARE ORDERED
*
      IF ( .NOT. QFIRST ) GO TO 140
      QFIRST = .FALSE.
      DO 100 M = 1, NPDAYS
      STORE(M) = DPRINT(M)
 100  CONTINUE
      NPDM = NPDAYS - 1
      DO 120 N = 1, NPDM
      K = N + 1
      DO 110 M = K, NPDAYS
      IF ( STORE(N) .LT. STORE(M) ) GO TO 110
      SAVE = STORE(N)
      STORE(N) = STORE(M)
      STORE(M) = SAVE
 110  CONTINUE
 120  CONTINUE
      KOUNT = 0
      DO 130 M = 1, NPDAYS
      IF ( STORE(M) .LT. START .OR.
     *     STORE(M) .GT. FINISH  .OR.
     *     STORE(M) .EQ. STORE(M - 1 ) ) GO TO 130
      KOUNT = KOUNT + 1
      DPRINT(KOUNT) = STORE(M)
 130  CONTINUE
 140  CONTINUE
*
*  DETERMINE IF HEADING IS NEEDED
```

```
*
      IF ( DAY .GT. START ) GO TO 150
      KOUNT = 1
      QHEAD = .TRUE.
      GO TO 160
 150  CONTINUE
      QHEAD = LINES .GT. MAXLIN .OR. QPDAY
 160  CONTINUE
      QPDAY = DAY .EQ. DPRINT(KOUNT)
      IF ( .NOT. QHEAD ) GO TO 170
*
*  PRINT HEADING
*
      LINES = 0
      WRITE ( LFILE, 510 ) TITLE
      IF ( YEAR(1) .GT. 0 ) WRITE ( LFILE, 520 ) YEAR
      WRITE ( LFILE, 530 )
      IF ( .NOT. QNKWAL ) WRITE ( LFILE, 540 )
      IF ( .NOT. QVERI ) WRITE ( LFILE, 560 )
      WRITE ( LFILE, 570 )
      IF ( .NOT. QNKWAL ) WRITE ( LFILE, 580 )
      IF ( .NOT. QVERI ) WRITE ( LFILE, 590 )
      WRITE ( LFILE, 600 )
      IF ( .NOT. QNKWAL ) WRITE ( LFILE, 610 )
      IF ( .NOT. QVERI ) WRITE ( LFILE, 620 )
      WRITE ( LFILE, 500 )
 170  CONTINUE
*
*  PRINT SUMMARY INFORMATION
*
      LINES = LINES + 1
      OUTFLO = SUMOUT
      ELEV = BOTTOM + DEPTH
      NOPN = OPEN
      IF ( OPEN .GT. 3 ) NOPN = 3
      WRITE ( LFILE, 630 ) DAY, ELEV, TEMPIN(1),
     *                     FLOWIN(1), AVTEMP,  OUTFLO,
     *                     ( PORT(K), PHLOW(K), K = 1, NOPN )
      IF ( .NOT. QNKWAL ) WRITE ( LFILE, 640 )
     *                   QUALIN(1, 1), AVQUAL(1)
      IF ( .NOT. QVERI ) WRITE ( LFILE, 650 ) TARGET
*
*  ADDITIONAL INFLOW POINTS
*
      IF ( NIP .LE. 1 ) GO TO 190
      LINES = LINES + NIP
      DO 180 L = 2, NIP
      WRITE ( LFILE, 660 ) TEMPIN(L), FLOWIN(L)
      IF ( .NOT. QNKWAL ) WRITE ( LFILE, 670 ) QUALIN(L, 1)
 180  CONTINUE
      WRITE ( LFILE, 500 )
 190  CONTINUE
*
*  PRINT PROFILES FOR SPECIAL PRINT DAYS
*
      IF ( .NOT. QPDAY ) GO TO 200
      KOUNT = KOUNT + 1
      CALL DETAIL
      CALL LINPLT
      CALL XGRAPH
 200  CONTINUE
      RETURN
 500      FORMAT ( / )
 501      FORMAT ( 5X, I3 )
 502      FORMAT ( 20F5.1 )
 510      FORMAT ( '1' // 20X, A )
 520      FORMAT ( 20X, I4, 'H - HYDROLOGY' /
     *             20X, I4, 'H - METEOROLOGY' )
 530      FORMAT ( // 2X, 'DAY', 3X, 'POOL', 4X, 'INFLOW',
     *            13X, 'INFLOW', 14X, 'RELEASE', 16X, 'OUTFLOW' )
 540      FORMAT ( '+', 25X, 'INFLOW', 34X, 'RELEASE' )
 560      FORMAT ( '+', 46X, 'TARGET' )
 570      FORMAT ( 8X, 'ELEV', 5X, 'TEMP', 13X,
     *             'QUANTITY', 15X, 'TEMP',17X,
     *             'QUANTITY', 3 (4X, 'PORT - FLOW' ) )
 580      FORMAT ( '+', 25X, 'QUAL-1', 35X, 'QUAL-1' )
 590      FORMAT ( '+', 47X, 'TEMP' )
 600      FORMAT ( 8X, '(FT)', 3X, '(DEG-C)', 14X, '(CFS)',
     *            14X, '(DEG-C)', 17X, '(CFS)', 1X,
     *             3 ( 10X, '(CFS)' ) )
 610      FORMAT ( '+', 25X, '(MG/L)', 35X, '(MG/L)' )
 620      FORMAT ( '+', 45X, '(DEG-C)' )
 630      FORMAT ( 2X, I3, 2X, F6.1, 2X, F6.2, 14X, F6.0,
     *            14X, F6.2, 17X, F6.0, 1X,
     *            3 ( 4X, I2, 3X, F6.0 ) )
 640      FORMAT ( '+', 25X, F5.1, 36X, F5.1 )
 650      FORMAT ( '+', 45X, F6.2 )
 660      FORMAT ( 15X, F6.2, 14X, F6.0 )
 670      FORMAT ( '+', 25X, F5.1 )
          END
```

```
      SUBROUTINE XREAD
*
*  THIS SUBROUTINE READS INPUT DATA FOR SIMULATION CONTROL
*  AND RESERVOIR PHYSICAL AND OPERATIONAL CHARACTERISTICS
*
      COMMON / AA / LSURF, NQUAL, QUAL(3, 100), DENC(3), SETV(3), THEAT
      COMMON / BB / TEMP(100), HGT(100), VOLUME(100), TARGET, TMIX, STAB
      COMMON / CH / NM, TITLE
      COMMON / DD / DELZ, BOTTOM, VEL(100), TEMPRT(10)
      COMMON / EE / KFILE, LFILE, JFILE, IFILE
      COMMON / FF / NPORTS, PAREA(8), PHGT(8)
      COMMON / GG / FLOWIN(3), TEMPIN(3), QUALIN(3, 3)
      COMMON / HH / DEPTH, ODEPTH, MAXLAY, NIP, WTHDRW(100)
      COMMON / II / EK, ET, SHORT, BETA, LAMBDA, MIXCOEF, GAMMA, DECAY
      COMMON / JJ / SUMOUT, WIDTH(100)
      COMMON / KK / NWELL(8), FMAX(8), FMIN(8), SELMAX
      COMMON / LL / FGAREA, FGHGT, FGMAX, FGMIN, FGANG, LPORT(8)
      COMMON / PP / FIRST, LAST, NJ, INDEX(366)
      COMMON / QQ / QVERI, QOUTC, QINTC, QINCFS, QOCFS, QWCFS
      COMMON / SS / OUTFLO(366, 8), TARG(366)
      COMMON / UU / ENFLOW(100), SUMFLO(366), DPRINT(14)
      COMMON / VV / START, FINISH, QRELE, QPROF
      COMMON / WW / QNKWAL, YEAR(2)
      COMMON / XX / QJUNK, ONE, TWO
      COMMON / ZZ / DINIT, TINIT(100), QINIT(3, 100)
      COMMON / BC / WRLNG, WRHGT, WRTYPE, DCOEF
      COMMON / CD / QPORT, QWEIR, QVINCR, QINITC
      COMMON / IJ / WANGLE, WTHETA(8)
      CHARACTER TITLE*78, NM*3, DUMMY(20)*4, NAME(2)*4
      REAL      LAMBDA, MIXCOEF
      INTEGER   FIRST, START, FINISH, ONE, TWO, DPRINT
      LOGICAL*4 QVERI, QFIRST, QOUTC, QMORE, QWCFS,
     *          QJUNK, QINTC, QINCFS, QOCFS, QECHO,
     *          QINITC, QVINCR, QNKWAL,
     *          QPORT, QWEIR, QRELE, QPROF, QCHECK
      CHARACTER*4 XFILE, XSTOP, XPLOT, XSETV, CHECK,
     *            TYPE, WRTYPE, UNITS,
     *            XPRED, XSINC, XVERI, XQUAL, XINFL,
     *            XLAYE, XTHIC, XBOTT, XVOLU, XWIDT, XPORT,
     *            XAREA, XHEIG, XMINI, XMAXI, XWETW, XSELM,
     *            XFLOO, XHEAT, XENTR, XDENC, XINTE, XANGL,
     *            XSIMU, XPRIN, XTARG, XFAHR, BLANK, XDEPT,
     *            XTEMP, XCELS, XINCR, XWEIR, XFREE, XSUBM,
     *            XRELE, XPROF, XCOEF, XEQUI, XMIXI
      DATA IFILE / 5 /
      DATA XFILE, XSTOP / 'FILE', 'STOP' /
      DATA XVERI, XPRED, XSINC / 'VERI', 'PRED', 'SINC' /
      DATA XINFL, XQUAL / 'INFL', 'QUAL' /
      DATA XLAYE, XTHIC, XBOTT / 'LAYE', 'THIC', 'BOTT' /
      DATA XVOLU, XWIDT, XPORT / 'VOLU', 'WIDT', 'PORT' /
      DATA XAREA, XHEIG, XMINI / 'AREA', 'HEIG', 'MINI' /
      DATA XMAXI, XWETW, XSELM / 'MAXI', 'WETW', 'SELM' /
      DATA XFLOO, XHEAT, XANGL / 'FLOO', 'HEAT', 'ANGL' /
      DATA XENTR, XDENC, XINTE / 'ENTR', 'DENC', 'INTE' /
      DATA XSIMU, XPRIN, XTARG / 'SIMU', 'PRIN', 'TARG' /
      DATA XPLOT, XFAHR, BLANK / 'PLOT', 'FAHR', '    ' /
      DATA XTEMP, XCELS, XINCR / 'TEMP', 'CELS', 'INCR' /
      DATA XWEIR, XFREE, XSUBM / 'WEIR', 'FREE', 'SUBM' /
      DATA XRELE, XPROF, XCOEF / 'RELE', 'PROF', 'COEF' /
      DATA XDEPT, XEQUI, XMIXI / 'DEPT', 'EQUI', 'MIXI' /
      DATA XSETV               / 'SETV'                 /
      DATA QFIRST, QMORE / 2 * .TRUE. /
      IF ( .NOT. QFIRST ) GO TO 360
      QFIRST = .FALSE.
*
*  PROGRAM CONTROL PARAMETERS
*
      READ (IFILE,530) TITLE
      READ (IFILE, 500 ) CHECK, IFILE, LFILE, JFILE, KFILE
      IF ( CHECK .NE. XFILE ) CALL ERROR ( 1000 )
      IF ( JFILE .LE. 0 .OR. JFILE .GT. 99 ) JFILE = IFILE
      IF ( KFILE .LE. 0 .OR. KFILE .GT. 99 ) KFILE = LFILE
      READ ( IFILE, 500 ) CHECK
      QECHO = CHECK .EQ. XPRIN
      IF ( .NOT. QECHO ) GO TO 130
*
*  ECHO PRINT INPUT
*
      QECHO = .FALSE.
      IF ( JFILE .EQ. IFILE ) NFILES = 1
      IF ( JFILE .NE. IFILE ) NFILES = 2
      DO 120 I = 1, NFILES
        IF ( I .EQ. 1 ) MFILE = IFILE
        IF ( I .EQ. 2 ) MFILE = JFILE
        REWIND MFILE
        LINE = 1000
 100  CONTINUE
      IF ( MOD ( ( LINE - 1000 ) / 10, 54 ) .EQ. 0 )  WRITE (LFILE, 550)
      READ ( MFILE, 530 ) DUMMY
      IF ( DUMMY(1) .EQ. XSTOP ) GO TO 110
      WRITE ( LFILE, 560 ) LINE, DUMMY
      LINE = LINE + 10
      GO TO 100
 110  CONTINUE
      REWIND MFILE
      WRITE ( LFILE, 560 ) LINE, XSTOP
 120  CONTINUE
      READ ( IFILE, 530 ) DUMMY
      READ ( IFILE, 530 ) DUMMY
      READ ( IFILE, 530 ) DUMMY
 130  CONTINUE
*
*  PROGRAM CONSTANTS
*
      READ ( IFILE, 580 ) CHECK, NAME
      IF ( CHECK .NE. XPLOT ) GO TO 150
      DO 140 M = 1, 2
        IF ( NAME(M) .EQ. XRELE ) QRELE = .TRUE.
        IF ( NAME(M) .EQ. XPROF ) QPROF = .TRUE.
 140  CONTINUE
 150  CONTINUE
      IF ( CHECK .EQ. XVERI .OR. CHECK .EQ. XPRED )  BACKSPACE IFILE
      READ ( IFILE, 580 ) CHECK, NAME
      QVERI = CHECK .EQ. XVERI
      IF ( CHECK .NE. XVERI .AND. CHECK .NE. XPRED ) CALL ERROR ( 1010 )
      READ ( IFILE, 570 ) CHECK, NQUAL, NAME
      QNKWAL = (CHECK .EQ. XQUAL .AND. NQUAL .LE. 0).OR.CHECK .NE. XQUAL
      IF ( CHECK .NE. XQUAL ) THEN
        NQUAL = 0
        BACKSPACE IFILE
      END IF
      READ ( IFILE, 500 ) CHECK, NIP
      IF ( CHECK .NE. XINFL .AND. CHECK .NE. XLAYE ) CALL ERROR ( 1020 )
      IF ( CHECK .EQ. XINFL ) GO TO 160
      NIP = 1
      BACKSPACE IFILE
 160  CONTINUE
      READ ( IFILE, 500 ) CHECK, MAXLAY
      IF ( CHECK .NE. XLAYE ) CALL ERROR ( 1030 )
      READ ( IFILE, 510 ) CHECK, DELZ
      IF ( CHECK .NE. XTHIC ) CALL ERROR ( 1040 )
*
*  RESERVOIR GEOMETRY
*
      READ ( IFILE, 510 ) CHECK, BOTTOM
      IF ( CHECK .NE. XBOTT ) CALL ERROR ( 1050 )
      READ ( IFILE, 540 ) CHECK, UNITS, TYPE
      IF ( CHECK .NE. XVOLU ) CALL ERROR ( 1060 )
      QVINCR = TYPE .EQ. XINCR
      READ ( IFILE, 520 ) ( VOLUME(I), I=1, MAXLAY )
      READ ( IFILE, 500 ) CHECK
      IF ( CHECK .NE. XWIDT ) CALL ERROR ( 1070 )
      READ ( IFILE, 520 ) ( WIDTH(I), I = 1, MAXLAY )
*
*  DESCRIPTION OF WITHDRAWAL DEVICE
*
      READ ( IFILE, 500 ) CHECK, NPORTS
      IF ( CHECK .NE. XPORT .AND. CHECK .NE. XWEIR ) CALL ERROR (1080)
      QPORT = CHECK .EQ. XPORT
      QWEIR = CHECK .EQ. XWEIR
      IF ( QPORT ) GO TO 170
      IF ( QWEIR ) GO TO 190
 170  CONTINUE
*
*  PORT
*
      READ ( IFILE, 510 ) CHECK, ( PAREA(K), K = 1, NPORTS )
      IF ( CHECK .NE. XAREA ) CALL ERROR ( 1090 )
      READ ( IFILE, 510 ) CHECK, ( WTHETA(K), K = 1, NPORTS )
      IF ( CHECK .NE. XANGL ) CALL ERROR ( 1091 )
      READ ( IFILE, 510 ) CHECK, ( PHGT(K), K = 1, NPORTS )
      IF ( CHECK .NE. XHEIG ) CALL ERROR ( 1100 )
      IF ( QVERI ) GO TO 180
      READ ( IFILE, 510 ) CHECK, ( FMIN(K), K = 1, NPORTS )
      IF ( CHECK .NE. XMINI ) CALL ERROR ( 1110 )
      READ ( IFILE, 510 ) CHECK, ( FMAX(K), K = 1, NPORTS )
      IF ( CHECK .NE. XMAXI ) CALL ERROR ( 1120 )
      READ ( IFILE, 500 ) CHECK, ( NWELL(K), K = 1, NPORTS )
      IF ( CHECK .NE. XWETW ) CALL ERROR ( 1130 )
      READ ( IFILE, 510 ) CHECK, SELMAX
      IF ( CHECK .NE. XSELM ) CALL ERROR ( 1140 )
      READ ( IFILE, 510 ) CHECK, FGAREA, FGHGT, FGMIN, FGMAX, FGANG
      IF ( CHECK .NE. XFLOO ) CALL ERROR ( 1150 )
 180  CONTINUE
*
*  WEIR
*
      READ ( IFILE, 510 ) CHECK, BETA, LAMBDA
      IF ( CHECK .NE. XWEIR .AND. CHECK .NE. XHEAT ) CALL ERROR ( 1160 )
      IF ( CHECK .EQ. XHEAT ) GO TO 200
      QWEIR = CHECK .EQ. XWEIR
 190  CONTINUE
      READ ( IFILE, 500 ) WRTYPE
      IF ( WRTYPE .NE. XFREE .AND. WRTYPE .NE. XSUBM ) CALL ERROR (1170)
      READ ( IFILE, 510 ) CHECK, WRLNG
      READ ( IFILE, 510 ) CHECK, WRHGT
      IF ( WRTYPE .NE. XFREE ) GO TO 200
      READ ( IFILE, 510 ) CHECK, DCOEF
      IF ( CHECK .NE. XCOEF ) CALL ERROR ( 1180 )
*
*  INTERNAL PROCESSES
*
      READ ( IFILE, 510 ) CHECK, BETA, LAMBDA
      IF ( CHECK .NE. XHEAT ) CALL ERROR ( 1190 )
 200  CONTINUE
      READ ( IFILE, 510 ) CHECK, MIXCOEF, DECAY
      IF ( CHECK .NE. XMIXI ) CALL ERROR ( 1195 )
      READ ( IFILE, 510 ) CHECK, GAMMA
      IF ( CHECK .NE. XENTR ) CALL ERROR ( 1197 )
      READ ( IFILE, 510 ) CHECK, ( DENC(J), J = 1, NQUAL )
      IF ( CHECK .NE. XDENC ) THEN
        DO 210 J = 1, NQUAL
        DENC(J) = 0.0
 210    CONTINUE
        BACKSPACE IFILE
      END IF
      READ ( IFILE, 510 ) CHECK, ( SETV(J), J = 1, NQUAL )
      IF ( CHECK .NE. XSETV ) THEN
        DO 220 J = 1, NQUAL
        SETV(J) = 0.0
 220    CONTINUE
        BACKSPACE IFILE
      END IF
*
*  SIMULATION CONTROL DATA
*
      READ ( IFILE, 500 ) CHECK, FIRST, LAST
      IF ( CHECK .NE. XINTE ) CALL ERROR ( 1240 )
      READ ( IFILE, 500 ) CHECK, START, FINISH, ONE, TWO
      IF ( CHECK .NE. XSIMU ) CALL ERROR ( 1250 )
      READ ( IFILE, 500 ) CHECK, DPRINT
      IF ( CHECK .NE. XPRIN ) CALL ERROR ( 1260 )
      IF ( QVERI ) GO TO 250
*
*  TARGET TEMPERATURES
*
      READ ( IFILE, 540 ) CHECK, UNITS, TYPE
      IF ( CHECK .NE. XTARG ) CALL ERROR ( 1270 )
      IF (UNITS.NE.XFAHR .AND. UNITS .NE. XCELS .AND. UNITS .NE. BLANK)
     *  CALL ERROR( 1280 )
      QOUTC = .NOT. UNITS .EQ. XFAHR
      IF ( TYPE .NE. XSINC ) GO TO 240
      READ ( IFILE, 520 ) A, B, C, D
      DO 230 N = FIRST, LAST
        TARG(N) = A * SIN ( B * N + C ) + D
 230  CONTINUE
      GO TO 250
 240  CONTINUE
      READ ( IFILE, 520 ) ( TARG(N), N = FIRST, LAST )
 250  CONTINUE
*
*  INITIAL VALUES
*
      IF ( .NOT. QMORE ) GO TO 310
      DSAVE = DINIT
      READ ( IFILE, 510 ) CHECK, DINIT
      IF (CHECK .NE. XDEPT .AND. CHECK .NE. XSTOP .AND. CHECK.NE.XEQUI)
     *   CALL ERROR ( 1290 )
```

APPENDIX B: DEFINITION OF VARIABLES BY SUBPROGRAM

PROGRAM WESTEX

VARIABLE	TYPE	DIMENSION	DESCRIPTION
AVGQ	R	(366,3)	DOWNSTREAM RELEASE QUALITY FOR EACH QUALITY PARAMETER, MG/L
AVGT	R	(366)	DOWNSTREAM RELEASE TEMPERATURE, DEG-C
AVQUAL	R	(3)	DOWNSTREAM RELEASE QUALITY FOR EACH QUALITY PARAMETER, MG/L
AVTEMP	R		DOWNSTREAM RELEASE TEMPERATURE, DEG-C
C1	R		FOUR CONSTANTS USED IN DETERMINING
C2	R		DENSITY CORRESPONDING TO A
C3	R		PARTICULAR TEMPERTURE
C4	R		
DELZ	R		THICKNESS OF A RESERVOIR LAYER, FT
DENC	R	(3)	SPECIFIC GRAVITY OF QUALITY CONSTITUENTS
DINIT	R		INITIAL VALUE OF POOL DEPTH, FT
DENSQ	R		INCREMENTAL DENSITY DUE TO QUALITIES
DENST	R		INCREMENTAL DENSITY DUE TO TEMPERATURE
DEPTH	R		DEPTH OF POOL, FT
DPRINT	I	(14)	SPECIFIED DAYS FOR WHICH DETAILED SIMULATION OUTPUT IS DESIRED
EK	R		EXCHANGE COEFFICIENT, BTU/SQ FT/DAY/ DEG-F
EQTEMP	R	(366)	EQUILIBRIUM TEMPERATUE, FOR EACH DAY, DEG-F
ET	R		EQUILIBRIUM TEMPERATURE, DEG-F
EXCOEF	R	(366)	EXCHANGE COEFFICIENT FOR EACH DAY, BTU/SQ FT/DAY/DEG-F
FINISH	I		LAST DAY OF SIMULATION
FLOWIN	R	(3)	INFLOW QUANTITY FOR EACH INFLOW POINT
HGT	R	(100)	PERCENTAGE AT WHICH EACH LAYER IS FILLED WITH WATER
I	I		INDEX FOR LAYERS
IJK	I		INDEX OF OPEN PORTS

PROGRAM WESTEX

VARIABLE	TYPE	DIMENSION	DESCRIPTION
INFLO	R	(366, 3)	INFLOW QUANTITY FOR EACH DAY FOR EACH INFLOW POINT
INQUAL	R	(366, 3, 3)	INFLOW QUALITY FOR EACH DAY FOR EACH INFLOW POINT FOR EACH QUALITY
INTEMP	R	(366, 3)	INFLOW TEMPERATURE FOR EACH DAY FOR EACH INFLOW POINT
J	I		INDEX FOR QUALITIES
K	I		INDEX FOR PORTS
KFILE	I		FILE CODE FOR OUTPUT OF INFORMATION FOR DEBUGGING
L	I		INDEX FOR INFLOW POINTS
LSURF	I		CURRENT NUMBER OF LAYERS
MAXLAY	I		MAXIMUM NUMBER OF LAYERS IN RESERVOIR
N	I		INDEX FOR DAYS
NFLOOD	I		NUMBER OF THE PORT REPRESENTING THE FLOOD GATE
NIP	I		NUMBER OF INFLOW POINTS
NPORTS	I		NUMBER OF SELECTIVE WITHDRAWAL PORTS NOT INCLUDING FLOOD GATE
NQUAL	I		NUMBER OF QUALITIES
ODEPTH	R		DEPTH OF POOL, FT, AT THE BEGINNING OF THE SIMULATION
ONE	I		FIRST DAY OF SPECIAL PRINTOUT FOR DEBUGGING
OPEN	I		NUMBER OF OPEN PORTS
OUTFLO	R	(366, 8)	OUTFLOW RATE FOR EACH DAY FOR EACH PORT, CFS
PHLOW	R	(3)	OUTFLOW THROUGH EACH OPEN PORT
POOL	R		POOL ELEVATION AT END OF SIMULATION DAY
PORT	I	(3)	PORT NUMBER OF EACH OPEN PORT
QFIRST	L		.TRUE. - FIRST CYCLE THRU THIS SECTION OF CODE - USED TO INITIALIZE PARAMETERS
QINIT	R	(3, 100)	INITIAL PROFILE OF QUALITIES
QJUNK	L		.TRUE. - INFLOW SPECIAL DEBUGGING OUTPUT IS REQUESTED FOR CURRENT SIMULATION DAY

PROGRAM WESTEX

VARIABLE	TYPE	DIMENSION	DESCRIPTION
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QPORT	L		.TRUE. - PORTS ARE PRESENT AS AN OUTLET DEVICE .FALSE. - NO PORTS
QPRINT	L		.TRUE. - OUTPUT WILL BE PRINTED .FALSE. - ONLY OUTPUT WILL BE PLOTS
QUAL	R	(3, 100)	QUALITY PROFILES
QUALIN	R	(3, 3)	INFLOW QUALITY FOR EACH INFLOW POINT FOR EACH QUALITY PARAMETER
QVERI	L		.TRUE. - VERIFICATION MODE .FALSE. - PREDICTION MODE
QWEIR	L		.TRUE. - A WEIR IS INCLUDED AS AN OUTPUT DEVICE .FALSE. - A WEIR IS NOT PRESENT
SHORT	R		SHORT WAVE SOLAR RADIATION
SOLAR	R	(366)	SHORT WAVE SOLAR RADIATION FOR EACH DAY
START	I		FIRST DAY OF SIMULATION
SUMFLO	R	(366)	TOTAL OUTFLOW FOR EACH DAY, CFS
SUMOUT	R		TOTAL OUTFLOW, CFS
TARG	R	(366)	TARGET TEMPERATURES FOR EACH DAY
TARGET	R		TARGET TEMPERATURE
TEMP	R	(100)	TEMPERATURE PROFILE
TEMPIN	R	(3)	INFLOW TEMPERATURE FOR EACH INFLOW POINT
TINIT	R	(100)	INITIAL TEMPERATURE PROFILE
TWO	I		LAST DAY OF SPECIAL PRINTOUT FOR DEBUGGING
WRFLO	R	(366)	OUTFLOW QUANTITY OVER WEIR FOR EACH DAY
WRFLOW	R		OUTFLOW QUANTITY OVER WEIR

SUBROUTINE CONVRT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
ENFLOW	R	(100)	TOTAL INFLOW PROFILE
FACTOR	R		CONVERSION FROM K-ACRE FT TO CFS
FGANG	R		WITHDRAWAL ANGLE FOR FLOODGATE
FGAREA	R		AREA OF FLOOD GATE, SQ FT
FGHGT	R		HEIGHT FROM BOTTOM OF FLOOD GATE CENTERLINE, FT
FGMAX	R		MAXIMUM FLOOD GATE CAPACITY, CFS
FGMIN	R		MINIMUM FLOOD GATE CAPACITY, CFS
FIRST	I		FIRST DAY OF DATA INPUT
FLOWIN	R	(3)	INFLOW QUANTITIES FOR ONE DAY
FMAX	R	(8)	MAXIMUM FLOW THRU PORT, CFS
FMIN	R	(8)	MINIMUM FLOW THRU PORT, CFS
I	I		INDEX FOR LAYERS
INFLO	R	(366, 3)	INFLOW QUANTITY FOR ONE INFLOW POINT
INTEMP	R	(366, 3)	INFLOW TEMPERATURE FOR EACH DAY FOR EACH INFLOW POINT
K	I		INDEX FOR OPEN PORTS
L	I		INDEX FOR INFLOW POINTS
LAST	I		LAST DAY OF DATA INPUT
LSURF	I		CURRENT NUMBER OF LAYERS
MAX	I		MAXIMUM NUMBER OF LAYERS - 1
MAXLAY	I		MAXIMUM NUMBER OF LAYERS IN RESERVOIR
N	I		INDEX FOR DAYS
NFLOOD	I		NUMBER OF THE PORT REPRESENTING THE FLOOD GATE
NIP	I		NUMBER OF INFLOW POINTS
NPORTS	I		NUMBER OF SELECTIVE WITHDRAWAL PORTS NOT INCLUDING FLOODGATE
OPEN	I		NUMBER OF OPEN PORTS
OUTFLO	R	(366, 8)	OUTFLOW RATE FOR EACH DAY FOR EACH PORT, CFS
PAREA	R	(8)	PORT AREA, SQ FT
PHGT	R	(8)	HEIGHT FROM BOTTOM OF PORT CENTERLINE

SUBROUTINE CONVRT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
PHLOW	R	(3)	OUTFLOW THROUGH EACH OPEN PORT
QFIRST	L		.TRUE. - FIRST CYCLE THRU THIS SECTION OF CODE - USED TO INITIALIZE PARAMETERS
QINCFS	L		.TRUE. - INFLOW QUANTITIES ARE IN UNITS OF CUBIC FEET PER SECOND
QINITC	L		.TRUE. - INITIAL TEMPERATURE PROFILE IN UNITS OF DEGREES CELSIUS
QINTC	L		.TRUE. - INFLOW TEMPERATURES ARE IN UNITS OF DEGREES CELSIUS
QOCFS	L		.TRUE. - OUTFLOW QUANTITIES ARE IN UNITS OF CFS
QOUTC	L		.TRUE. - TARGET TEMPERATURES ARE INPUT IN UNITS OF DEGREES CELSIUS
QPORT	L		.TRUE. - PORTS ARE PRESENT AS AN OUTLET DEVICE .FALSE. - NO PORTS
QVERI	L		.TRUE. - VERIFICATION MODE .FALSE. - PREDICTION MODE
QVINCR	L		.TRUE. - RESERVOIR VOLUMES ARE INPUT AS INCREMENTAL VOLUMES FOR EACH LAYER .FALSE. - RESERVOIR VOLUMES ARE INPUT AS CUMULATIVE VOLUMES AT THE TOP OF EACH LAYER
QWCFS	L		.TRUE. - WEIR FLOW QUANTITIES ARE IN CFS
QWEIR	L		.TRUE. - A WEIR IS INCLUDED AS AN OUTPUT DEVICE
SELMAX	R		MAXIMUM CAPACITY FOR SELECTIVE WITHDRAWAL SYSTEM, CFS
SMALL	R		ESSENTIALLY ZERO
SUMFLO	R	(366)	TOTAL OUTFLOW FOR EACH DAY, CFS
SUMOUT	R		TOTAL OUTFLOW, CFS
TARG	R	(366)	TARGET TEMPERATURES FOR EACH DAY
TINIT	R	(100)	INITIAL TEMPERATURE PROFILE
VOLUME	R	(100)	VOLUME OF RESERVOIR LAYERS IN 1000 ACRE-FT
WRFLO	R	(366)	OUTFLOW QUANTITY OVER WEIR FOR EACH DAY
WTHDRW	R	(100)	WITHDRAWAL PROFILE
WTHETA	R	(8)	WITHDRAWAL ANGLE FOR THE PORTS

SUBROUTINE DECIDE

VARIABLE	TYPE	DIMENSION	DESCRIPTION
DAY	I		CURRENT SIMULATION DAY
DELZ	R		THICKNESS OF A VERTICAL LAYER, FT
EXTRA	R		FLOW VOLUME NOT RELEASED AS SCHEDULED BUT INSTEAD SAVED UNTIL THE NEXT DAY
F1	R		FLOW FROM ONE OF TWO OPEN PORTS BEING USED FOR BLENDING
F2	R		FLOW FROM ONE OF TWO OPEN PORTS BEING USED FOR BLENDING
FGMIN	R		MINIMUM FLOW THRU FLOODGATE, CFS
FLOW	R		FLOW SCHEDULED FOR RELEASE FOR CURRENT SIMULATION DAY
FLW	R		FLOW ALLOCATED TO A SELECTIVE WITHDRAWAL PORT
FMAX	R	(8)	MAXIMUM FLOW THRU PORT, CFS
FMIN	R	(8)	MINIMUM FLOW THRU PORT, CFS
FSLMT	R		FLOW REMAINING TO BE ALLOCATED BEFORE SELECTIVE WITHDRAWAL SYSTEM CAPACITY LIMIT IS EXCEEDED
I	I		INDEX FOR LAYERS - TEMPORARY INDEX FOR PORTS
IJK	I		INDEX OF OPEN PORTS
INDEX	I	(366)	PORT SELECTION INDEX - INDICATES IN WHICH SECTION OF CODE PORT SELECTION DECISION WAS MADE FOR EACH DAY
J	I		TEMPORARY INDEX FOR PORTS
K	I		INDEX FOR PORTS
K1	I		TEMPORARY INDEX FOR PORTS
L	I		INDEX FOR WETWELLS
LPORT	I	(8)	LAYER CONTAINING EACH PORT
LSURF	I		CURRENT NUMBER OF LAYERS
M	I		PORT SELECTION INDEX
N	I		TEMPORARY VARIABLE FOR SORTING PORT ELEVATIONS
NFLOOD	I		NUMBER OF THE PORT REPRESENTING THE FLOOD GATE
NPORTS	I		NUMBER OF SELECTIVE WITHDRAWAL PORTS NOT INCLUDING FLOOD GATE

SUBROUTINE DECIDE

VARIABLE	TYPE	DIMENSION	DESCRIPTION
NWELL	I	(8)	IDENTIFICATION OF WHICH WETWELL CONTAINS A SPECIFIC PORT
ODEPTH	R		DEPTH OF POOL, FT, AT THE BEGINNING OF THE SIMULATION
OPEN	I		NUMBER OF OPEN PORTS
PAREA	R	(8)	PORT AREA, SQ FT
PHGT	R	(8)	HEIGHT FROM BOTTOM OF PORT CENTERLINE
PHLO	R		TEMPORARY VARIABLE - FLOW ALLOCATED TO A PORT
PHLOW	R	(3)	OUTFLOW THROUGH EACH OPEN PORT
PORT	I	(3)	PORT NUMBER OF EACH OPEN PORT
Q1	L		USED TO IDENTIFY WETWELL NUMBER
Q2	L		USED TO IDENTIFY WETWELL NUMBER
QCHECK	L		.TRUE. PORTS ARE ORDERED FROM TOP TO BOTTOM
QD	L	(8)	.TRUE. - ELEVATION OF PORT IS LESS THAN WATER SURFACE ELEVATION
QFIRST	L		.TRUE. - FIRST CYCLE THRU THIS SECTION OF CODE - USED TO INITIALIZE PARAMETERS
QMORE	L		.TRUE. - CURRENT OPERATION STRATEGY WILL BE USED WITH SELECTIVE WITHDRAWAL FOR UPDATED OPERATION STRATEGY
QWELL	L	(8)	USED TO IDENTIFY WETWELL NUMBER FOR EACH PORT
REST	R		FLOW REMAINING TO BE ALLOCATED
SCALE	R		USED TO SCALE WITHDRAWAL PROFILE SUCH THAT TOTAL WITHDRAWAL IS ACCOUNTED FOR
SELMAX	R		MAXIMUM CAPACITY FOR SELECTIVE WITHDRAWAL SYSTE, CFS
SMALL	R		ESSENTIAL ZERO
SUM	R		SUM OF WITHDRAWAL QUANTITIES FROM EACH LAYER
SUMOUT	R		SCHEDULED OUTFLOW TO BE RELEASED, CFS
SUMTF	R		SUM OVER LAYERS OF PRODUCT OF WITHDRAWAL AND TEMPERATURE
TARGET	R		TARGET TEMPERATURE
TEMP	R	(100)	TEMPERATURE PROFILE

SUBROUTINE DECIDE

VARIABLE	TYPE	DIMENSION	DESCRIPTION
TNEW	R		UPDATED REFERENCE TEMPERATURE FOR A PORT
TPORT	R	(8)	REFERENCE TEMPERATURE FOR A PORT - INITIALLY CENTERLINE TEMPERATURE OF A PORT
TX	R		TARGET TEMPERATURE FOR A SINGLE PORT WHEN ANOTHER PORT IS CONSTRAINED TO BE OPEN
VEL	R	(100)	VELOCITY PROFILE
WIDTH	R	(100)	WIDTH OF RESERVOIR LAYERS, FT
WTHDRW	R	(100)	WITHDRAWAL PROFILE
X	R		TEMPORARY VARIABLE FOR SORTING PORT ELEVATIONS
XFLOW	R		TEMPORARY VARIABLE FOR FLOW THROUGH EACH OPEN PORT
XOPEN	I		TEMPORARY VARIABLE FOR NUMBER OF OPEN PORTS
XPORT	I		TEMPORARY VARIABLE INDICATING WHICH PORTS ARE OPEN
Y	R		TEMPORARY VARIABLE FOR SORTING PORT ELEVATIONS

FUNCTION DENINT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
DELZ	R		THICKNESS OF A VERTICAL LAYER, FT
DEN	R	(100)	DENSITY OF EACH LAYER, G/ML
DENINT	R		INTERPOLATED DENSITY
DENPRT	R		DENSITY AT PORT LOCATION
DEPTH	R		DEPTH OF POOL, FT
DGRD	R		DENSITY GRADIENT
DGRB	R		DENSITY GRADIENT BETWEEN THE PORT LOCATION AND BOTTOM
DGRT	R		DENSITY GRADIENT BETWEEN THE PORT LOCATION AND TOP OF THE POOL
DIFF	R		ABSOLUTE DIFFERENCE BETWEEN LOCATION AT WHICH DENSITY IS TO BE DETERMINED AND NEAREST LAYER MIDPOINT
ELMID	R		LOCATION OF MIDPOINT OF LAYER CONTAINING LOCATION AT WHICH DENSITY IS TO BE DETERMINED
ELTOP	R		ELEVATION OF MIDPOINT OF UPPER INTERPOLATION LAYER
HGTPRT	R		CENTERLINE ELEVATION FOR ONE OPEN PORT
IJ	I		UPPER INTERPOLATION LAYER
IJK	I		ZERO OR ONE - USED TO DEFINE INTERPOLATION LAYERS
JK	I		LOWER INTERPOLATION LAYER
LAYER	I		INDEX FOR A LAYER
LSURF	I		CURRENT NUMBER OF LAYERS
SIGN	R		+1 IF INTERPOLATION LOCATION IS BELOW MIDPOINT OF ITS LAYER -1 IF INTERPOLATION LOCATION IS ABOVE MIDPOINT OF ITS LAYER
SLOPE	R		CHANGE IN DENSITY BETWEEN TWO INTERPOLATION LAYERS
SMALL	R		ESSENTIALLY ZERO - A CHECK FOR A CONSTANT DENSITY CONDITION
X	R		LOCATION AT WHICH DENSITY IS REQUIRED

SUBROUTINE DETAIL

VARIABLE	TYPE	DIMENSION	DESCRIPTION
AVQUAL	R	(3)	DOWNSTREAM RELEASE QUALITY FOR EACH QUALITY PARAMETER, MG/L
AVTEMP	R		DOWNSTREAM RELEASE TEMPERATURE, DEG-C
BLANK	C		" " BLANK SPACES
BOTTOM	R		ELEVATION OF BOTTOM OF RESERVOIR, FT
DAY	I		CURRENT SIMULATION DAY
DEEP	R		DEPTH FROM SURFACE OF MIDPOINT OF A LAYER
DELZ	R		THICKNESS OF A VERTICAL LAYER, FT
DEPTH	R		DEPTH OF POOL, FT
EK	R		EXCHANGE COEFFICIENT
ELEV	R		ELEVATION OF MIDPOINT OF A LAYER
ENFLOW	R	(100)	TOTAL INFLOW PROFILE
ET	R		EQUILIBRIUM TEMPERATURE, DEG-F
FLOWIN	R	(3)	INFLOW QUANTITIES FOR ONE DAY FOR EACH INFLOW POINT
HGT	R	(100)	PERCENTAGE AT WHICH LAYER IS FILLED WITH WATER
I	I		INDEX FOR LAYERS
IJK	I		PORT NUMBER OF OPEN PORTS
J	I		INDEX FOR QUALITIES
JKL	I		LAYER CONTAINING OPEN PORTS
K	I		INDEX FOR PORTS
L	I		INDEX FOR INFLOW POINTS
LEVEL	I	(100)	ALPHA LABEL FOR EACH LAYER IDENTIFYING OPEN PORTS
LFILE	I		FILE CODE FOR SIMULATION OUTPUT
LSURF	I		CURRENT NUMBER OF LAYERS
LSURFP1	I		CURRENT NUMBER OF LAYERS + 1
M	I		COUNTER FOR DAYS
MAXLAY	I		MAXIMUM NUMBER OF LAYERS IN RESERVOIR
MONTH	I	(12)	NAMES OF MONTHS
NIP	I		NUMBER OF INFLOW POINTS

SUBROUTINE DETAIL

VARIABLE	TYPE	DIMENSION	DESCRIPTION
NJ	I		NUMBER OF DAY OF MONTH FOR CURRENT SIMULATION DAY
NM	I		NUMBER OF MONTH FOR CURRENT SIMULATION DAY
NQUAL	I		NUMBER OF QUALITIES
NUMBER	I	(12)	NUMBER OF DAYS IN EACH MONTH
OPEN	I		NUMBER OF OPEN PORTS
PHGT	R	(8)	HEIGHT FROM BOTTOM OF PORT CENTERLINE
PHLOW	R	(3)	OUTFLOW THROUGH EACH OPEN PORT
PORT	I	(3)	PORT NUMBER OF EACH OPEN PORT
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QUAL	R	(3, 100)	QUALITY PROFILES
QUALIN	R	(3, 3)	INFLOW QUALITY FOR EACH INFLOW POINT FOR EACH QUALITY PARAMETER
QVERI	L		.TRUE. - VERIFICATION MODE .FALSE. - PREDICTION MODE
SHORT	R		SHORT WAVE SOLAR RADIATION
SUMOUT	R		TOTAL OUTFLOW, CFS
TARGET	R		TARGET TEMPERATURE
TEMP	R	(100)	TEMPERATURE PROFILE
TEMPIN	R	(3)	INFLOW TEMPERATURE FOR EACH INFLOW POINT
TITLE	C	(20)	ALHANUMERIC IDENTIFICATION OF PROJECT
VEL	R	(100)	RELATIVE VELOCITY PROFILE
WTHDRW	R	(100)	WITHDRAWAL PROFILE
XXXX	I		"XXXX"
YEAR	I	(2)	YEARS OF METEOROLOGIC DATA AND HYDROLOGIC DATA

SUBROUTINE ERROR

VARIABLE	TYPE	DIMENSION	DESCRIPTION
IJK	I		ERROR CODE
LFILE	I		FILE CODE FOR SIMULATION OUTPUT

SUBROUTINE FINAL

VARIABLE	TYPE	DIMENSION	DESCRIPTION
AADIFF	R		AVERAGE ABSOLUTE DIFFERENCE OF RELEASE TEMPERATURES AND TARGET TEMPERATURES
AVDIFF	R		AVERAGE DIFFERENCE OF RELEASE TEMPERATURES AND TARGET TEMPERATURES
AVGQ	R	(366, 3)	DOWNSTREAM RELEASE QUALITY FOR EACH DAY FOR EACH QUALITY, MG/L
AVGT	R	(366)	DOWNSTREAM RELEASE TEMPERATURE FOR EACH DAY, DEG-C
B	I	(10)	TEMPORARY VARIABLE CONTAINING DATA VALUES IN ONE LINE OF FINAL OUTPUT TABLES
DIFF	R	(366)	DIFFERENCE IN RELEASE TEMPERATURE AND TARGET TEMPERATURE FOR EACH DAY
FINISH	I		LAST DAY OF SIMULATION
I	I		INDEX FOR LAYERS
INCR	I		NUMBER OF DATA VALUES FOR EACH COLUMN OF FINAL OUTPUT TABLES
INDEX	I	(366)	PORT SELECTION INDEX
J	I		INDEX FOR QUALITIES
K	I		INDEX FOR COLUMN IN OUTPUT TABLE
L	I	(10)	TEMPORARY VARIABLE CONTAINING DAY NUMBERS IN ONE LINE OF FINAL OUTPUT TABLES
LFILE	I		FILE CODE FOR SIMULATION OUTPUT
M	I		ROWS IN TABLE OF MAXIMUM DIFFERENCE
MXDIFF	R		MAXIMUM DIFFERENCE IN RELEASE TEMPERATURE AND TARGET TEMPERATURE
MXGRAD	R		MAXIMUM ONE DAY RELEASE TEMPERATURE CHANGE
N	I		INDEX FOR DAYS
NQUAL	I		NUMBER OF QUALITIES
NUMBER	I		NUMBER OF SIMULATION DAYS
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QVERI	L		.TRUE. - VERIFICATION MODEL .FALSE. - PREDICTION MODE
SADIFF	R		SUM OF ABSOLUTE DIFFERENCES OF RELEASE TEMPERATURES AND TARGET TEMPERATURES

SUBROUTINE FINAL

VARIABLE	TYPE	DIMENSION	DESCRIPTION
SMDIFF	R		SUM OF DIFFERENCES OF RELEASE TEMPERATURES AND TARGET TEMPERATURES
SSDIFF	R		SUM OF SQUARED DIFFERENCES OF RELEASE TEMPERATURES AND TARGET TEMPERATURES
START	I		FIRST DAY OF SIMULATION
TARG	R	(366)	TARGET TEMPERATURES FOR EACH DAY
TITLE	C	(20)	ALPHANUMERIC IDENTIFICATION OF PROJECT
X	R	(10)	TEMPORARY VARIABLE CONTAINING DATA VALUES IN ONE LINE OF FINAL OUTPUT TABLES
YEAR	I	(2)	YEARS OF METEOROLOGIC DATA AND HYDROLOGIC DATA

SUBROUTINE HEATEX

VARIABLE	TYPE	DIMENSION	DESCRIPTION
BETA	R		PERCENTAGE OF SHORT WAVE RADIATION RETAINED IN TOP TWO FEET OF POOL
BNEXT	R		DISTANCE FROM SURFACE TO THE BOTTOM OF THE LOWEST LAYER CONTAINING THE PENETRATION DEPTH
CHANGE	R		CHANGE OF TEMPERATURE IN A LAYER DUE TO SURFACE HEAT EXCHANGE
DAY	I		CURRENT SIMULATION DAY
DELZ	R		LAYER THICKNESS, FT
DEPTH	R		DEPTH OF POOL, FT
EK	R		SURFACE HEAT EXCHANGE COEFFICIENT
ET	R		EQUILIBRIUM TEMPERATURE, DEG-F
EXTRA	R		CONTRIBUTION OF SHORT WAVE RADIATION BETWEEN A TWO FOOT DEPTH AND THE LOWST LAYER OF SURFACE PENETRATION DEPTH
HDEPTH	R		SURFACE HEAT PENETRATION DEPTH - TWO FEET
HDOWN	R		SHORT WAVE DISTRIBUTED INTO ALL LAYERS BELOW LAYERS OF SURFACE HEAT PENETRATION DEPTH
HEAT	R	(100)	HEAT ENTERING EACH LAYER DUE TO SURFACE HEAT EXCHANGE
HGT	R	(100)	PERCENTAGE AT WHICH LAYER IS FILLED WITH WATER
HSM1	R		HEAT ENTERING LAYER BELOW SURFACE LAYER DUE TO SURFACE HEAT
HSM2	R		HEAT ENTERING LAYER TWO LAYERS BELOW SURFACE LAYER DUE TO SURFACE HEAT
HSURF	R		HEAT ENTERING SURFACE LAYER
HTOTAL	R		TOTAL SURFCE HEAT EXCHANGE
I	I		INDEX FOR LAYERS
KFILE	I		FILE CODE FOR OUTPUT OF INFORMATION FOR DEBUGGING
LAMBDA	R		LIGHT EXTINCTION COEFFICIENT - SHAPING COEFFICIENT FOR EXPONENTIAL DISTRIBUTION OF SHORT WAVE RADIATION, 1/FT
LSM	I		LOWEST LAYER TO WHICH SURFACE PENETRATION DEPTH EXTENDS
LSURF	I		CURRENT NUMBER OF LAYERS

SUBROUTINE HEATEX

VARIABLE	TYPE	DIMENSION	DESCRIPTION
QFEET	L		SURFACE HEAT PENETRATION DEPTH IS EXPRESSED IN FEET
QFIRST	L		.TRUE. - FIRST CYCLE THRU THIS SECTION OF CODE - USED TO INITIALIZE PARAMETERS
QJUNK	L		.TRUE. - SPECIAL DEBUGGING OUTPUT IS REQUESTED FOR CURRENT SIMULATION DAY
QMETR	L		SURFACE HEAT PENETRATION DEPTH IS EXPRESSED IN METRES
RHO	R		DENSITY OF WATER, 62.4 LBS/CU FT
SCALE	R		SCALING PARAMETER TO INSURE TOTAL HEAT ENTERS THE POOL
SHORT	R		SHORT WAVE SOLAR RADIATION
SMALL	R		ESSENTIALLY ZERO
SUM	R		SUM OF HEAT ENTERING EACH LAYER
TEMP	R	(100)	TEMPERATURE PROFILE
THETA	R		SURFACE TEMPERATURE, DEG-F
TOP	R		THICKNESS OF SURFACE LAYER
ZMID	R		DEPTH OF MIDPOINT OF LAYER FROM SURFACE

SUBROUTINE HMREAD

VARIABLE	TYPE	DIMENSION	DESCRIPTION
A	R		AMPLITUDE FOR HARMONIC RELATION A * SIN (B * T + C) + D USED FOR INPUT OF TARGET TEMPERATURES AND INFLOW TEMPERATURES
B	R		FREQUENCY FOR HARMONIC RELATION A * SIN (B * T + C) + D USED FOR INPUT OF TARGET TEMPERATURES AND INFLOW TEMPERATURES; CONVERSION FROM DAYS TO RADIANS (B = .0172)
BLANK	C		" " BLANK SPACES
C	R		PHASE SHIFT IN RADIANS FOR HARMONIC RELATION A * SIN (B * T + C) + D USED FOR INPUT OF TARGET TEMPERATURES AND INFLOW TEMPERATURES
CHECK	I		FIRST FOUR ALPHANUMERIC CHARACTERS OF A DATA INPUT CARD IMAGE
D	R		MEAN VALUE FOR HARMONIC RELATION A * SIN (B * T + C) + D
DFIRST	I		FIRST CALENDAR DAY OF A MONTH
DLAST	I		LAST CALENDAR DAY OF A MONTH
EQTEMP	R	(366)	EQUILIBRIUM TEMPERATUE, FOR EACH DAY, DEG-F
EXCOEF	R	(366)	EXCHANGE COEFFICIENT FOR EACH DAY, BTU/SQ FT/DAY/DEG-F
FIRST	I		FIRST DAY OF DATA INPUT
INFLO	R	(366, 3)	INFLOW QUANTITY FOR EACH DAY FOR EACH INFLOW POINT
INQUAL	R	(366, 3, 3)	INFLOW QUALITY FOR EACH DAY FOR EACH INFLOW POINT FOR EACH QUALITY
INTEMP	R	(366, 3)	INFLOW QUANTITY FOR EACH DAY FOR EACH INFLOW POINT
J	I		INDEX FOR QUALITIES
JFILE	I		FILE CODE FOR INPUT OF HYDROMETEOROLOGICAL DATA
K	I		INDEX FOR PORTS
L	I		INDEX FOR INFLOW POINTS
LAST	I		LAST DAY OF DATA INPUT
M	I		INDEX FOR MONTHS
MFIRST	I		FIRST MONTH OF MONTHLY INPUT DATA
MLAST	I		LAST MONTH OF MONTHLY INPUT DATA

SUBROUTINE HMREAD

VARIABLE	TYPE	DIMENSION	DESCRIPTION
N	I		INDEX FOR DAYS
NIP	I		NUMBER OF INFLOW POINTS
NPORTS	I		NUMBER OF SELECTIVE WITHDRAWAL PORTS NOT INCLUDING FLOODGATE
NQUAL	I		NUMBER OF QUALITIES
NUMBER	I	(13)	CALENDAR DAY OF BEGINNING OF EACH MONTH
OUTFLO	R	(366, 8)	OUTFLOW RATE FOR EACH DAY FOR EACH PORT, CFS
QFIRST	L		.TRUE. - FIRST CYCLE THRU THIS SECTION OF CODE - USED TO INITIALIZE PARAMETERS
QINCFS	L		.TRUE. - INFLOW QUANTITIES ARE UNITS OF CUBIC FEET PER SECOND
QINTC	L		.TRUE. - INFLOW TEMPERATURES ARE IN DEGREES CELSIUS
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QOCFS	L		.TRUE. - OUTFLOW QUANTITIES ARE IN UNITS OF CFS
QPORT	L		.TRUE. - PORTS ARE PRESENT AS AN OUTLET DEVICE .FALSE. - NO PORTS
QVERI	L		.TRUE. - VERIFICATION MODE .FALSE. - PREDICTION MODE
QWEIR	L		.TRUE. - A WEIR IS INCLUDED AS AN OUTLET DEVICE
SOLAR	R	(366)	SOLAR RADIATION FOR EACH DAY
SUMFLO	R	(366)	TOTAL OUTFLOW FOR EACH DAY, CFS
TYPE	I		INDICATES SPECIAL TYPE OF INPUT (CUMULATIVE RESERVOIR VOLUME OR SINE CURVE FOR INPUT TEMPERATURES OF MONTHLY INPUT)
UNITS	I		UNITS OF INPUT DATA
VALUE	R		DATA VALUES FOR MONTHLY INPUT
WIND	R	(366)	WIND SPEED, MPH
WRFLO	R	(366)	OUTFLOW QUANTITY OVER WEIR FOR EACH DAY
XCELS	C		"CELS"IUS
XCFS	C		"CFS"

SUBROUTINE HMREAD

VARIABLE	TYPE	DIMENSION	DESCRIPTION
XDEPT	C		"DEPT"H
XEND	C		"END"
XEQUI	C		"EQUI"LIBRIUM TEMPERATURE
XEXCH	C		"EXCH"ANGE COEFFICIENT
XFAHR	C		"FAHR"ENHEIT
XINFL	C		"INFL"OW
XKACF	C		"K-AC"RE FT
XMONT	C		"MONT"HLY
XOUTF	C		"OUTF"LOWS
XQUAL	C		"QUAL"ITIES
XSHOR	C		"SHOR"T WAVE RADIATION
XSINC	C		"SINC"URVE
XSTOP	C		"STOP"
XTEMP	C		"TEMP"ERATURE
XWEIR	C		"WEIR"
XWIND	C		"WIND"
XYZ	R		TEMPORARY VARIABLE FOR MONTHLY INPUT DATA
YEAR	I	(2)	YEARS OF METEOROLOGIC DATA AND HYDROLOGIC DATA

SUBROUTINE INFLOW

VARIABLE	TYPE	DIMENSION	DESCRIPTION
C1	R		FOUR CONSTANTS USED IN DETERMINING
C2	R		DENSITY CORRESPONDING TO A
C3	R		PARTICULAR TEMPERTURE
C4	R		
DAY	I		CURRENT SIMULATION DAY
DELZ	R		THICKNESS OF A RESERVOIR LAYER, FT
DEN	R	(100)	DENSITY OF EACH LAYER, G/ML
DENC	R	(3)	SPECIFIC GRAVITY OF QUALITY CONSTITUENTS
DENMIX	R		DENSITY OF MIXED VOLUME OF INFLOW AND ENTRAINMENT VOLUME
DENSQ	R		INCREMENTAL DENSITY DUE TO QUALITIES
DENST	R		INCREMENTAL DENSITY DUE TO TEMPERATURE
DEPTH	R		DEPTH OF POOL, FT
ENFLOW	R	(100)	TOTAL INFLOW PROFILE
ENTFLO	R		VOLUME OF ENTRAINED WATER
FLOW	R		TEMPORARY VARIABLE USED IN DETERMINING THE NUMBER OF SURFACE LAYERS REQUIRED FOR ENTRAINMENT
FLOWIN	R	(3)	INFLOW QUANTITIES FOR EACH INFLOW POINT
GAMMA	R		ENTRAINMENT COEFFICIENT - PERCENTAGE OF INFLOW VOLUME ENTRAINED FROM SURFACE
HGT	R	(100)	PERCENTAGE AT WHICH LAYER IS FILLED WITH WATER
I	I		INDEX FOR LAYERS
INFLAY	I		LAYER INTO WHICH TOTAL INFLOW VOLUME ENTERS OR IS CENTERED
INFLO	R		INFLOW QUANTITY FOR ONE INFLOW POINT
INFLOE	R		INFLOW QUANTITY INTO A SPECIFIC LAYER
INLAY	R	(100)	INFLOW PROFILE FOR ONE INFLOW POINT
J	I		INDEX FOR QUALITIES
K	I		INDEX FOR LAYERS (FROM TOP TO BOTTOM)
KFILE	I		FILE CODE FOR SPECIAL DEBUGGING OUTPUT
L	I		INDEX FOR INFLOW POINTS

SUBROUTINE INFLOW

VARIABLE	TYPE	DIMENSION	DESCRIPTION
LSURF	I		CURRENT NUMBER OF LAYERS
MAXLAY	I		MAXIMUM NUMBER OF LAYERS
NIP	I		NUMBER OF INFLOW POINTS
NQUAL	I		NUMBER OF QUALITIES
QJUNK	L		.TRUE. - SPECIAL DEBUGGING OUTPUT IS REQUESTED FOR CURRENT SIMULATION DAY
QMIX	R	(3)	QUALITY OF MIXED VOLUME OF INFLOW AND ENTRAINMENT VOLUME
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QUAL	R	(3, 100)	QUALITY PROFILES
QUALIN	R	(3, 3)	INFLOW QUALITY FOR CURRENT SIMULATION DAY
QVMIX	R	(3)	PRODUCT OF QUALITY AND VOLUME IN MIXED INFLOW QUANTITY
SUMFL1	R		SUM OF INFLOW VOLUME AND ENTRAINMENT VOLUME
SUMVOL	R		SUM OF EXISTING VOLUME OF A LAYER, VOLUME ENTERING LAYER FROM LAYER BELOW AND INFLOW
TEMP	R	(100)	TEMPERATURE PROFILE
TEMPIN	R	(3)	INFLOW TEMPERATURES FOR CURRENT SIMULATION DAY
TMIX	R		TEMPERATURE OF MIXED VOLUME OF INFLOW AND ENTRAINMENT VOLUME
TVMIX	R		PRODUCT OF TEMPERATURE AND VOLUME OF MIXED INFLOW QUANTITY
UPFLOW	R		VOLUME DISPLACED UPWARD DUE TO INFLOW
UPQUAL	R	(3)	QUALITY OF VOLUME DISPLACED UPWARD DUE TO INFLOW
UPTEMP	R		TEMPERATURE OF VOLUME DISPLACED UPWARD DUE TO INFLOW
UPVOL	R		DISPLACED VOLUME DUE TO INFLOW EXCEEDING ACTUAL VOLUME OF CURRENT LAYER
VOLHGT	R		PRODUCT OF VOLUME AND PERCENTAGE TO WHICH CURRENT LAYER IS FILLED
VOLUME	R	(100)	VOLUME OF LAYERS

SUBROUTINE LINPLT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
BLANK	C		" " BLANK SPACES
BOTTOM	R		ELEVATION OF BOTTOM OF RESERVOIR, FT
CHANGE	I		CHANGE IN VALUE BETWEEN TIC MARK ON TEMPERATURE OR QUALITY AXES
COLUMN	C	(100)	CONTENTS FOR A ROW OR LAYER OF PROFILE PLOTS - MOSTLY BLANK
CONCOL	R	(3)	QUALITY CONCENTRATION OF A LAYER
CSPACE	I	(3, 11)	VALUE OF QUALITY AT EACH OF 11 TIC MARKS FOR EACH QUALITY
DAY	R		CURRENT SIMULATION DAY
DEEP	R		DEPTH BELOW SURFACE OF MIDPOINT OF A LAYER
DELZ	R		THICKNESS OF A VERTICAL LAYER, FT
DEPTH	R		DEPTH OF POOL, FT
ELEV	R		ELEVATION OF MIDPOINT OF A LAYER
I	I		INDEX FOR LAYERS
IJK	I		COLUMN THAT A TEMPERATURE OR QUALITY VALUE WILL BE PLOTTED IN
J	I		INDEX FOR QUALITIES
K	I		INDEX FOR LAYER FROM TOP TO BOTTOM
L	I		INDEX FOR TIC MARKS ON AXES OF PLOT
LFILE	I		FILE CODE FOR SIMULATION OUTPUT
LSURF	I		CURRENT NUMBER OF LAYERS
MAXQ	I	(3)	MAXIMUM VALUE FOR EACH QUALITY ON PLOT AXIS
MAXT	I		MAXIMUM VALUE OF TEMPERATURE ON PLOT AXIS
NJ	I		NUMBER OF DAY OF THE MONTH
NM	C		NUMBER OF THE MONTH
NQUAL	I		NUMBER OF QUALITIES
P	C	(3)	PLOT SYMBOL FOR EACH QUALITY
PEGGED	C		"*" - INDICATES DATA VALUE IS OUTSIDE RANGE OF PLOT
PLUS	C		"+"

SUBROUTINE LINPLT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QRANGE	L		.TRUE. - DATA VALUE FOR A LAYER IS WITHIN SPECIFIED RANGE OF PLOT
QUAL	R	(3, 100)	QUALITY PROFILES
T	C		"T" - PLOT SYMBOL FOR TEMPERATURE
TEMP	R	(100)	TEMPERATURE PROFILE
TITLE	C	(20)	ALHANUMERIC IDENTIFICATION OF PROJECT
TMPCOL	I		TEMPERATURE OF A LAYER
TSPACE	I	(11)	TEMPERATURE VALUE AT EACH TIC MARK ON PLOT AXIS
X	C		"X" - PLOT SYMBOL WHEN TWO PROFILES COINCIDE
YEAR	I	(2)	YEARS OF METEOROLOGIC DATA AND HYDROLOGIC DATA

SUBROUTINE MIXING

VARIABLE	TYPE	DIMENSION	DESCRIPTION
AIRRHO	R		DENSITY OF AIR, G/L
AREA	R	(100)	AREA OF THE LAYERS
C1	R		FOUR CONSTANTS USED IN DETERMINING
C2	R		DENSITY CORRESPONDING TO A
C3	R		PARTICULAR TEMPERATURE
C4	R		
C10	R		WIND STRESS COEFFICIENT
CCOEF	R		COEFFICIENT OF TURBULENT KINETIC ENERGY
CDRAG	R		WIND STRESS COEFFICIENT
CMIX	R		EDDY DIFFUSIVITY
CWD	R		COEFFICIENT FOR TOTAL KINETIC ENERGY, WIND GENERATED
DAY	I		CURRENT SIMULATION DAY
DECAY	R		COEFFICIENT FOR EDDY DIFFUSIVITY
DELQ	R	(3,100)	CHANGE IN QUALITY AS A RESULT OF MIXING BELOW THE MIXED LAYER
DELRHO	R		DENSITY DIFFERENCE BETWEEN MIXED LAYER AND ADJACENT LAYER
DELT	R	(3,100)	CHANGE IN TEMPERATURE AS A RESULT OF MIXING BELOW MIXED LAYER
DELVOL	R		VOLUME IN LAYER ADJACENT TO CURRENT MIXING LAYER
DELZ	R		THICKNESS OF A VERTICAL LAYER, FT
DELZONE	R		INCREMENT OF MIXED LAYER DEPTH
DEN	R	(100)	DENSITY OF EACH LAYER, G/ML
DENC	R	(3)	SPECIFIC GRAVITY OF QUALITY CONSTITUENTS
DENSQ	R		INCREMENTAL DENSITY DUE TO QUALITIES
DEPTH	R		DEPTH OF POOL, FT
DISSIP	R		DISSIPATION OF TOTAL TUBULENT KINETIC ENERGY
EK	R		EXCHANGE COEFFICIENT, BTU/SQ FT/DAY/ DEG-F
ET	R		EQUILIBRIUM TEMPERATURE, DEG-F
EXPAND	R		COEFFICIENT OF THERMAL EXPANSION

SUBROUTINE MIXING

VARIABLE	TYPE	DIMENSION	DESCRIPTION
FRI	R		RATIO OF TOTAL POTENTIAL ENERGY TO TOTAL CHANGE IN KINETIC ENERGY
G	R		ACCELERATION OF GRAVITY
HEAT	R		HEAT TRANSFER INTO THE SURFACE LAYER
HGT	R	(100)	PERCENTAGE AT WHICH LAYER IS FILLED WITH WATER
I	I		INDEX FOR LAYERS
J	I		INDEX FOR QUALITIES
KFILE	I		FILE CODE FOR INPUT OF HYDROMETEOROLOGICAL DATA
LFILE	I		FILE CODE FOR SIMULATION OUTPUT
LSM	I		NUMBER OF LAYERS - 1
LSM1	I		CURRENT NUMBER OF LAYERS
LSURF	I		CURRENT NUMBER OF LAYERS
M	I		INDEX FOR LAYERS
MAXLAY	I		MAXIMUM NUMBER OF LAYERS IN RESERVOIR
MIXCOEF	R		MIXING COEFFICIENT USED TO DETERMINE EDDY DIFFUSIVITIES
NQUAL	I		NUMBER OF QUALITIES
QFIRST	L		.TRUE. - FIRST CYCLE THRU THIS SECTION OF CODE - USED TO CALCULATE AREA OF THE LAYERS
QHELP	L		.TRUE. - SPECIAL DEBUGGING OUTPUT REQUESTED
QJUNK	L		.TRUE. - SPECIAL DEBUGGING OUTPUT IS REQUESTED FOR CURRENT SIMULATION DAY
QMIX	R	(3)	QUALITY OF MIXED LAYERS
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QUAL	R	(3, 100)	QUALITY PROFILES
RHOMIX	R		DENSITY OF MIXED LAYER
RI	R		RICHARDSON NUMBER
SHEAR	R		SURFACE SHEAR STRESS, WIND GENERATED
SPHEAT	R		SPECIFIC HEAT OF WATER, CAL/DEG-F/G
SUMVOL	R		TOTAL VOLUME IN MIXED LAYER

SUBROUTINE MIXING

VARIABLE	TYPE	DIMENSION	DESCRIPTION
SUMWORK	R		TOTAL WORK REQUIRED FOR MIXED LAYER TO EXTEND TO CURRENT LAYER
TEMP	R	(100)	TEMPERATURE PROFILE
TEMPMIX	R		TEMPERATURE OF MIXED LAYERS
THETA	R		SURFACE TEMPERATURE, DEG-C
TKE	R		TOTAL TURBULENT KINETIC ENERGY
TKECONV	R		TOTAL TURBULENT KINETIC ENERGY DUE TO OVERTURNING CONVECTION
TKEVECT	R		TOTAL TURBULENT KINETIC ENERGY DUE TO OVERTURNING CONVECTION
TKEWIND	R		TOTAL KINETIC ENERGY FROM WIND SHEAR
TLAYER	I		TOP LAYER OR CURRENT NUMBER OF LAYERS
VOLUME	R	(100)	VOLUME OF RESERVOIR LAYERS
WCOEFF	R		COEFFICIENT FOR TOTAL KINETIC ENERGY, WIND GENERATED
WIND	R	(366)	WIND SPEED, MPH
WORK	R		WORK REQUIRED TO LIFT CURRENT LAYER M TO CENTER OF MASS OF MIXED LAYER REGION
WSP	R		WIND SPEED ALTERNATIVE UNITS, METERS PER SECOND
WSPEED	R		WIND SPEED FOR SPECIFIC SIMULATION DAY, MPH
WSTAR	R		SHEAR VELOCITY
ZGRAVITY	R		CENTER OF GRAVITY OF MIXED LAYER
ZONE	R		DEPTH OF MIXED LAYER

SUBROUTINE OUTVEL

VARIABLE	TYPE	DIMENSION	DESCRIPTION
AREA	R		AREA OF ONE OPEN PORT
DELZ	R		THICKNESS OF A VERTICAL LAYER, FT
DEPTH	R		DEPTH OF POOL, FT
FLORAT	R		FLOW THROUGH ONE OPEN PORT
FVOL	R		TEMPORARY VARIABLE FOR OUTFLOW VOLUME
HGT	R	(100)	PERCENTAGE AT WHICH LAYER IS FILLED WITH WATER
HGTPRT	R		CENTERLINE ELEVATION FOR ONE OPEN PORT
I	I		INDEX FOR LAYERS
IJK	I		PORT NUMBER OF AN OPEN PORT
K	I		INDEX FOR PORTS
LAYPRT	I		LAYER CONTAINING AN OPEN PORT
LL	I	(8)	LOWER LIMIT OF WITHDRAWAL ZONE FROM EACH OUTLET
LOWLIM	I		LOWER WITHDRAWAL LIMIT
LSURF	I		CURRENT NUMBER OF LAYERS
LT	I	(8)	UPPER LIMIT OF WITHDRAWAL ZONE FROM EACH OUTLET
NUSURF	I		NUMBER OF LAYERS AFTER WITHDRAWAL
OPEN	I		NUMBER OF OPEN PORTS
PAREA	R	(8)	PORT AREA, SQ FT
PHGT	R	(8)	HEIGHT FROM BOTTOM OF PORT CENTERLINE
PHLOW	R	(3)	OUTFLOW THROUGH EACH OPEN PORT
PORT	I	(3)	PORT NUMBER OF EACH OPEN PORT
QPORT	L		.TRUE. - PORTS ARE PRESENT AS AN OUTLET DEVICE .FALSE. - NO PORTS
QWEIR	L		.TRUE. - A WEIR IS INCLUDED AS AN OUTPUT DEVICE .FALSE. - NO PORTS
SCALE	R		COMPUTED FACTOR TO SCALE VELOCITY PROFILE
SUMOUT	R		TOTAL OUTFLOW, CFS

SUBROUTINE OUTVEL

VARIABLE	TYPE	DIMENSION	DESCRIPTION
SUMVW	R		SUM OF VELOCITIES FROM ALL LAYERS IN THE WITHDRAWAL ZONE
TOPLIM	I		UPPER WITHDRAWAL LIMIT
V	R	(100)	RELATIVE VELOCITY PROFILE FOR ONE PORT
VEL	R	(100)	TOTAL VELOCITY PROFILE
VMAX	R		RELATIVE MAXIMUM VELOCITY
VOLHGT	R		PRODUCT OF VOLUME AND PERCENTAGE TO WHICH CURRENT LAYER IS FILLED
VOLUME	R	(100)	VOLUME OF LAYERS
WANGLE	R		WITHDRAWAL ANGLE OF SPECIFIC OUTLET
WRFLOW	R		WEIR FLOW FOR CURRENT SIMULATION DAY
WTHETA	R	(8)	WITHDRAWAL ANGLE OF EACH OUTLET

SUBROUTINE REFILL

VARIABLE	TYPE	DIMENSION	DESCRIPTION
AVQUAL	R	(3)	DOWNSTREAM RELEASE QUALITY FOR EACH DAY
AVTEMP	R		DOWNSTREAM RELEASE TEMPERATURE, DEG-C
C1	R		FOUR CONSTANTS USED IN DETERMINING
C2	R		DENSITY CORRESPONDING TO A
C3	R		PARTICULAR TEMPERATURE
C4	R		
DAY	I		CURRENT SIMULATION DAY
DELZ	R		LAYER THICKNESS, FT
DEN	R	(100)	DENSITY OF EACH LAYER, G/ML
DENC	R	(3)	SPECIFIC GRAVITY OF QUALITY CONSTITUENTS
DENSQ	R		INCREMENTAL DENSITY DUE TO QUALITIES
DENST	R		INCREMENTAL DENSITY DUE TO TEMPERATURE
DEPTH	R		DEPTH OF POOL, FT
DOWN	R		FLOW VOLUME TRANSPORTED INTO A LAYER FROM THE LAYER ABOVE
EXTRA	R		FLOW VOLUME FROM AN UPPER LAYER OCCURRING WHEN WITHDRAWAL VOLUME FOR A LAYER EXCEEDS AVAILABLE VOLUME OF THE LAYER
FLOW	R		WITHDRAWAL VOLUME OF A LAYER
HGT	R	(100)	PERCENTAGE AT WHICH LAYER IS FILLED WITH WATER
I	I		INDEX FOR LAYERS
J	I		INDEX FOR QUALITIES
K	I		INDEX FOR PORTS
KFILE	I		FILE CODE FOR OUTPUT OF INFORMATION FOR DEBUGGING
ISM	I		NUMBER OF LAYERS - 1
LSURF	I		CURRENT NUMBER OF LAYERS
NQUAL	I		NUMBER OF QUALITIES
NUSP1	I		NUMBER OF THE SURFACE LAYER + 1
NUSURF	I		NUMBER OF LAYERS AFTER WITHDRAWAL
ODEPTH	I		DEPTH OF THE POOL

SUBROUTINE REFILL

VARIABLE	TYPE	DIMENSION	DESCRIPTION
QDOWN	R		.TRUE. - PRODUCT OF A QUALITY AND FLOW TRANSPORTED INTO A LOWER LAYER
QJUNK	L		.TRUE. - SPECIAL DE-BUGGING OUTPUT IS REQUESTED FOR CURRENT SIMULATION DAY
QMOVE	L		.TRUE. - NUMBER OF LAYERS DECREASES AS A RESULT OF WITHDRAWAL
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QREMV	R		PRODUCT OF A QUALITY AND VOLUME REMAINING IN A LAYER
QUAL	R	(3, 100)	QUALITY PROFILES
REMVOL	R		VOLUME REMAINING IN A LAYER
SCALE	R		SCALING PARAMETER TO INSURE TOTAL SPECIFIED VOLUME IS WITHDRAWN
SUM	R		SUM OF WITHDRAWAL QUANITIES FROM EACH LAYER
SUMF	R		SUM OVER LAYERS OF FLOW VOLUMES
SUMOUT	R		TOTAL OUTFLOW, CFS
SUMQF	R		SUM OVER LAYERS OF PRODUCT OF WITHDRAWAL AND QUALITY CONSTITUENT
SUMTF	R		SUM OVER LAYERS OF PRODUCT OF WITHDRAWAL AND TEMPERATURE
TDOWN	R		PRODUCT OF TEMPERATURE AND FLOW TRANSPORTED INTO A LOWER LAYER
TEMP	R	(100)	TEMPERATURE PROFILE
TFLOW	R		TOTAL WITHDRAWAL FLOW
TREMV	R		PRODUCT OF TEMPERATURE AND VOLUME REMAINING IN A LAYER
VEL	R	(100)	RELATIVE VELOCITY PROFILE
VOL	R		VOLUME OF A LAYER
VOLHGT	R		PRODUCT OF VOLUME AND PERCENTAGE TO WHICH CURRENT LAYER IS FILLED
VOLUME	R	(100)	VOLUME OF LAYERS
WTHDRW	R	(100)	WITHDRAWAL PROFILE

SUBROUTINE SETTLE

VARIABLE	TYPE	DIMENSION	DESCRIPTION
DELZ	R		THICKNESS OF A RESERVOIR LAYER, FT
HGT	R	(100)	PROPORTION OF LAYER FILLED WITH WATER
I	I		INDEX FOR LAYERS
J	I		INDEX FOR QUALITIES
K	I		LAYER NUMBER CONTAINING THE CONSTITUENT MASS TO BE SETTLED INTO LAYER I DURING THE PRESENT SIMULATION DAY
LOWER	I		LOWER BOUND OF CONTROL VOLUME AT THE BEGINNING OF THE SIMULATION DAY, FT
LSURF	I		SURFACE LAYER
NQUAL	I		NUMBER OF QUALITIES
QUAL	R	(3, 100)	QUALITY PROFILES
SETV	R	(3)	SETTLING VELOCITIES FOR QUALITY CONSTITUENTS, FT/DAY
SURF	R		WATER DEPTH IN THE RESERVOIR, FT
TOPL	R		HEIGHT OF WATER TO THE K-LAYER, FT
UPPER	R		UPPER BOUND OF THE CONTROL VOLUME AT THE BEGINNING OF THE SIMULATION DAY, FT

SUBROUTINE STABITY

VARIABLE	TYPE	DIMENSION	DESCRIPTION
CP	R		SPECIFIC HEAT CAPACITY OF WATER
D	R		DEPTH OF WATER IN THE RESERVOIR, FT
DDEL	R		DENSITY DIFFERENCE BETWEEN THE AVERAGE AND THAT IN EACH LAYER, G/ML
DELZ	R		THICKNESS OF A RESERVOIR LAYER, FT
DEN	R	(100)	DENSITY OF EACH LAYER, G/ML
DENA	R		AVERAGE WATER DENSITY IN THE RESERVOIR, G/ML
HGT	R	(100)	PROPORTION OF LAYER FILLED WITH WATER
I	I		INDEX FOR LAYERS
LSURF	I		SURFACE LAYER
PERT	R		PROPORTION OF THE SURFACE LAYER FILLED WITH WATER
SAREA	R		SURFACE AREA OF THE RESERVOIR, FT2
STOT	R		SUMMATION OF DENSITY DIFFERENCE TIMES MOMENT ARM, TIMES WATER VOLUME
TEMP	R	(100)	WATER TEMPERATURE OF LAYERS, DEG-C
THEAT	R		TOTAL HEAT IN THE RESERVOIR
TMASS	R		TOTAL MASS IN THE RESERVOIR, G AC-FT/ML
TVOL	R		TOTAL VOLUME OF WATER IN THE RESERVOIR IN 1000 AC-FT
VOL	R	(100)	VOLUME OF RESERVOIR LAYERS IN 1000 AC-FT
ZC	R		HEIGHT ABOVE BOTTOM AT WHICH THE WATER DENSITY EQUALS THE AVERAGE RESERVOIR WATER DENSITY, FT
ZDEL	R		DISTANCE BETWEEN THE LAYER OF AVERAGE DENSITY AND EACH OF THE OTHER LAYERS, FT

SUBROUTINE STABLE

VARIABLE	TYPE	DIMENSION	DESCRIPTION
C1	R		FOUR CONSTANTS USED IN DETERMINING
C2	R		DENSITY CORRESPONDING TO A
C3	R		PARTICULAR TEMPERATURE
C4	R		
DAY	I		CURRENT SIMULATION DAY
DEN	R	(100)	DENSITY OF EACH LAYER, G/ML
DENC	R	(3)	SPECIFIC GRAVITY OF QUALITY CONSTITUENTS
DENMIX	R		DENSITY OF MIXED LAYERS
DENSQ	R		INCREMENTAL DENSITY DUE TO QUALITIES
DENST	R		INCREMENTAL DENSITY DUE TO TEMPERATURE
HGT	R	(100)	PERCENTAGE AT WHICH LAYER IS FILLED WITH WATER
I	I		INDEX FOR LAYERS
J	I		INDEX FOR QUALITIES
K	I		INDEX FOR MIXED LAYERS
KFILE	I		FILE CODE FOR SPECIAL DEBUGGING OUTPUT
LSURF	I		CURRENT NUMBER OF LAYERS
MIXLOW	I		LOWER LAYER OF MIXED ZONE
MIXTOP	I		UPPER LAYER OF MIXED ZONE
NQUAL	I		NUMBER OF QUALITIES
QJUNK	L		.TRUE. - SPECIAL DEBUGGING OUTPUT IS REQUESTED FOR CURRENT SIMULATION DAY
QMIX	R	(3)	QUALITY OF MIXED LAYERS
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QUAL	R	(3, 100)	QUALITY PROFILES
SUMQV	R		SUM OVER MIXED LAYERS OF A QUALITY AND MIXED VOLUME
SUMTV	R		SUM OVER MIXED LAYERS OF TEMPERATURE AND MIXED LAYER
SUMVOL	R		SUM OF MIXED VOLUMES OVER MIXED LAYER
TEMP	R	(100)	TEMPERATURE PROFILE
TMIX	R		TEMPERATURE OF MIXED LAYERS

SUBROUTINE STABLE

VARIABLE	TYPE	DIMENSION	DESCRIPTION
VLMXLW	R		VOLUME OF THE LOWEST LAYER OF THE MIXED ZONE
VLMXTP	R		VOLUME OF THE HIGHEST LAYER OF THE MIXED ZONE
VOLHGT	R		PRODUCT OF VOLUME AND PERCENTAGE TO WHICH CURRENT LAYER IS FILLED
VOLUME	R	(100)	VOLUME OF LAYERS

SUBROUTINE TPLOT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
AVGQ	R	(366, 3)	DOWNSTREAM RELEASE QUALITY FOR EACH DAY FOR EACH QUALITY, MG/L
AVGT	R	(366)	DOWNSTREAM RELEASE TEMPERATURE FOR EACH DAY, DEG-C
DELZ	R		LAYER THICKNESS
DEPTH	R		DEPTH OF POOL, FT
ELEV	R		HEIGHT FROM BOTTOM OF MIDPOINT OF A LAYER
FINISH	I		LAST DAY OF SIMULATION
HMAX	R		MAXIMUM POOL DEPTH
I	I		INDEX FOR LAYERS
ISM	I		NUMBER OF LAYERS - 1
J	I		INDEX FOR QUALITIES
LSURF	I		CURRENT NUMBER OF LAYERS
MAXLAY	I		MAXIMUM NUMBER OF LAYERS
N	I		INDEX FOR DAYS
NJ	I		NUMBER OF DAY OF MONTH FOR CURRENT SIMULATION DAY
NM	I		NUMBER OF MONTH FOR CURRENT SIMULATION DAY
NQUAL	I		NUMBER OF QUALITIES
PFILE	I		OUTPUT FILE FOR PLOT INFORMATION
QFC	L		.TRUE. - FIRST TIME THRU SUBROUTINE
QFG	L		.TRUE. - FIRST TIME THRU PROFILE PLOTTING SECTION OF CODE
QFT	L		.TRUE. - FIRST TIME THRU SEASONAL PLOTTING SECTION OF CODE
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QUAL	R	(3, 100)	QUALITY PROFILES
START	I		FIRST DAY OF SIMULATION
TEMP	R	(100)	TEMPERATURE PROFILE
TITLE	C	(20)	ALPHANUMERIC IDENTIFICATION OF PROJECT
YEAR	I	(2)	YEARS OF METEOROLOGIC DATA AND HYDROLOGIC DATA

SUBROUTINE VOLUME

VARIABLE	TYPE	DIMENSION	DESCRIPTION
I	I		INDEX FOR LAYERS
ITM1	I		UPPER LAYER - 1
ITOP	I		LAYER CONTAINING THE WATER SURFACE
PER	R		PROPORTION THAT A LAYER IS FILLED WITH WATER
TVOL	R		TOTAL VOLUME OF WATER IN THE RESERVOIR IN 1000 AC-FT
VOL	R	(100)	VOLUME OF RESERVOIR LAYERS IN 1000 AC-FT

SUBROUTINE VPORT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
AREA	R		AREA OF PORT
AVGVEL	R		AVERAGE VELOCITY THRU PORT FPS
BONLIM	R		THE BOUNDARY LIMIT WITH EITHER SURFACE OR BOTTOM BOUNDARY INTERFERENCE
C2	R		ANGLE OF WITHDRAWAL DIVIDED BY PI (3.14159)
DELDEN	R		DENSITY DIFFERENCE FROM LAYER OF MAXIMUM VELOCITY TO LOCAL ELEVATION
DELZ	R		THICKNESS OF A VERTICAL LAYER, FT
DEN	R	(100)	DENSITY OF EACH LAYER, G/ML
DENBOT	R		DENSITY AT THE BOTTOM OF THE RESERVOIR
DENDIF	R		DENSITY DIFFERENCE FROM LAYER OF MAXIMUM VELOCITY TO A WITHDRAWAL LIMIT
DENLIM	R		DENSITY AT THE LIMIT OF WITHDRAWAL
DENLOW	R		DENSITY AT LOWER WITHDRAWAL LIMIT
DENPRT	R		DENSITY AT CENTERLINE OF PORT LOCATION
DENTOP	R		DENSITY AT UPPER WITHDRAWAL LIMIT
DENUPP	R		DENSITY AT THE TOP OF THE RESERVOIR
DEPTH	R		DEPTH OF POOL, FT
DRBLIM	R		DIFFERNCE IN DENSITY BETWEEN THE CENTERLINE OF THE PORT AND THE LOWER WITHDRAWAL LIMIT
DRPBOT	R		DIFFERENCE IN DENSITY BETWEEN THE CENTERLINE OF THE PORT AND THE BOTTOM OF THE PORT
DRPTOP	R		DIFFERENCE IN DENSITY BETWEEN THE CENTERLINE OF THE PORT AND THE TOP OF THE PORT
DRTLIM	R		DIFFERENCE IN DENSITY BETWEEN THE CENTERLINE OF THE PORT AND THE UPPER WITHDRAWAL LIMIT
DVMAX	R		DENSITY AT LOCATION OF MAXIMUM VELOCITY
F1	R		VALUE OF WITHDRAWAL LIMIT FUNCTION AT AN ELEVATION
F3	R		VALUE OF WITHDRAWAL LIMIT FUNCTION AT AN INTERMEDIATE ELEVATION
FLOCFS	R		FLOW RATE THRU PORT IN CFS

SUBROUTINE VPORT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
FLORAT	R		FLOW RATE THRUE PORT IN K-ACRE FT
G	R		GRAVITATIONAL ACCELERATION - 32.2 FT/SEC/SEC
H1	R		VALUE OF SMITH ET AL. WITHDRAWAL LIMIT FUNCTION AT AN ELEVATION
H3	R		VALUE OF SMITH ET AL. WITHDRAWAL LIMIT FUNCTION AT AN INTERMEDIATE ELEVATION
HGTLOW	R		HEIGHT ABOVE BOTTOM OF LOWER LIMIT
HGTPRT	R		HEIGHT ABOVE BOTTOM OF PORT CENTERLINE
HGTTOP	R		HEIGHT ABOVE BOTTOM OF UPPER LIMIT
I	I		INDEX FOR LAYERS
LOWLIM	I		LAYER OF LOWER LIMT
LVMAX	I		LAYER OF MAXIMUM VELOCITY
MAX	I		NUMBER OF SEARCH ITERATIONS
NUSURF	I		NUMBER OF LAYERS AFTER WITHDRAWAL
PHIFRAC	R		RATIO OF THE IN-POOL WITHDRAWAL ZONE TO THE TOTAL (THEORETICAL) WITHDRAWAL ZONE
PRTBOT	R		HEIGHT ABOVE BOTTOM OF THE PORT INVERT
PRTTOP	R		HEIGHT ABOVE BOTTOM OF THE TOP OF PORT OPENING
QBLIM	L		.TRUE. - BOTTOM WITHDRAWAL INTERFERENCE
QSHIFT	L		.TRUE. - CALL SUBROUTINE TO ADJUST MULTIPLE WITHDRAWAL ZONES TO ACCOUNT FOR OVERLAP
QSINK1	L		.TRUE. - LOWER WITHDRAWAL LIMIT DOES NOT VIOLATE THE POINT SINK ASSUMPTION
QSINK2	L		.TRUE. - UPPER WITHDRAWAL LIMIT DOES NOT VIOLATE THE POINT SINK ASSUMPTION
QTLIM	L		.TRUE. - SURFACE WITHDRAWAL INTERFERENCE
RATIO	R		RATIO OF PRODUCT OF LOCAL HEIGHT AND DENSITY DIFFERENCE TO PRODUCT OF WITHDRAWAL ZONE THICKNESS AND DENSITY DIFFERENCE
SINK1	R		RATIO OF THE PRODUCT OF DISTANCE FROM THE PORT CENTERLINE AND DENSITY DIFFERENCE FOR THE LOWER WITHDRAWAL LIMIT AND FOR THE PORT INVERT

SUBROUTINE VPORT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
SINK2	R		RATIO OF THE PRODUCT OF DISTANCE FROM THE PORT CENTERLINE AND DENSITY DIFFERENCE FOR THE UPPER WITHDRAWAL LIMIT AND FOR THE TOP OF THE PORT OPENING
SMALL	R		10% OF LAYER THICKNESS
SMALLB	R		DIFFERENCE IN ELEVATION BETWEEN THE LOCATION OF THE PORT AND THE SURFACE
TINY	R		ESSENTIALLY ZERO, 1.0 E - 08
TOPLIM	I		LAYER OF UPPER LIMIT
V	R	(100)	LOCAL VELOCITY PROFILE
VD2	R		ONE-HALF THE VERTICAL DIMENSION OF THE PORT ASSUMING A SQUARE OPENING
VDIM	R		VERTICAL DIMENSION OF THE PORT ASSUMING A SQUARE OPENING
VDIM2	R		ONE-HALF THE VERTICAL DIMENSION OF THE PORT ASSUMING A SQUARE OPENING
VMAX	R		MAXIMUM VELOCITY
WANGLE	R		WITHDRAWAL ANGLE OF A SPECIFIC PORT
X1	R		ELEVATION OF A SEARCH LIMIT
X2	R		ELEVATION OF A SEARCH LIMIT
X3	R		ELEVATION OF A NEW SEARCH LIMIT
X4	R		ELEVATION OF A PREVIOUS SEARCH LIMIT
XVMAX	R		LOCATION OF MAXIMUM VELOCITY RELATIVE TO BOTTOM
Y	R		DISTANCE FROM LOCATION OF MAXIMUM VELOCITY TO LOCAL ELEVATION
YVMAX	R		LOCATION OF MAXIMUM VELOCITY REFERENCED TO LOWER WITHDRAWAL LIMIT
ZLOW	R		DISTANCE BETWEEN PORT AND LOWER WITHDRAWAL LIMIT
ZONE	R		DISTANCE FROM LOWER WITHDRAWAL LIMIT TO UPPER WITHDRAWAL LIMIT
ZONED	R		DISTANCE FROM BOUNDARY OF INTERFERENCE TO THE FREE LIMIT OF WITHDRAWAL
ZTOP	R		DISTANCE BETWEEN PORT AND UPPER WITHDRAWAL LIMIT

SUBROUTINE VWEIR

VARIABLE	TYPE	DIMENSION	DESCRIPTION
A	R		COEFFICIENT FOR X = A + B * CD FOR DETERMINING AN EXPONENT AS A FUNCTION OF DISCHARGE COEFFICIENT
AVGVEL	R		AVERAGE VELOCITY THRU PORT, FPS
B	R		COEFFICIENT FOR X = A + B * CD FOR DETERMINING AN EXPONENT AS A FUNCTION OF DISCHARGE COEFFICIENT
CREST	R		WEIR CREST HEIGHT FROM BOTTOM
DCOEF	R		DISCHARGE COEFFICIENT
DELDEN	R		DENSITY DIFFERENCE FROM LAYER OF MAXIMUM VELOCITY TO LOCAL ELEVATION
DELZ	R		LAYER THICKNESS, FT
DEN	R	(100)	DENSITY OF EACH LAYER, G/ML
DENDIF	R		DENSITY DIFFERENCE FROM LAYER OF MAXIMUM VELOCITY TO A WITHDRAWAL LIMIT
DENLOW	R		DENSITY AT LOWER WITHDRAWAL LIMIT
DENTOP	R		DENSITY AT UPPER WITHDRAWAL LIMIT
DEPTH	R		DEPTH OF POOL, FT
DVMAX	R		DENSITY AT LOCATION OF MAXIMUM VELOCITY
EXPNT	R		EXPONENT FOR VELOCITY PROFILE FOR FREE WEIR FLOW
F1	R		VALUE OF WITHDRAWAL LIMIT FUNCTION AT AN EXTREME LOCATION
F2	R		VALUE OF WITHDRAWAL LIMIT FUNCTION AT AN EXTREME LOCATION
F3	R		VALUE OF WITHDRAWAL LIMIT FUNCTION AT AN INTERMEDIATE ELEVATION
FLOCFS	R		FLOW RATE THRU PORT IN CFS
FLORAT	R		FLOW RATE THRU PORT IN K-ACRE-FT
G	R		GRAVITATIONAL ACCELERATION
HEAD	R		HEAD OVER WEIR
HGTLOW	R		HEIGHT OF LOWER LIMIT ABOVE BOTTOM
HGTTOP	R		HEIGHT OF UPPER LIMIT ABOVE BOTTOM
I	I		INDEX FOR LAYERS
ITMAX	I		NUMBER OF SEARCH ITERATIONS

SUBROUTINE VWEIR

VARIABLE	TYPE	DIMENSION	DESCRIPTION
LOWLIM	I		LAYER OF LOWER LIMIT
LVMAX	I		LAYER OF MAXIMUM VELOCITY
NUSURF	I		NUMBER OF LAYERS AFTER WITHDRAWAL
P	R	(3)	EXPONENT FOR VELOCITY PROFILE FUNCTION
Q1	L		.TRUE. - POSITIVE WITHDRAWAL LIMIT FUNCTION
Q2	L		.TRUE. - POSITIVE WITHDRAWAL LIMIT FUNCTION
QBLIM	L		.TRUE. - BOTTOM WITHDRAWAL INTERFERENCE
QFIRST	L		.TRUE. - FIRST CYCLE THRU THIS SECTION OF CODE - USED TO INITIALIZE PARAMETERS
QTLIM	L		.TRUE. - SURFACE WITHDRAWAL INTERFERENCE
RATIO	R		RATIO OF PRODUCT OF LOCAL HEIGHT AND DENSITY DIFFERENCE TO PRODUCT OF WITHDRAWAL ZONE THICKNESS AND DENSITY DIFFERENCE
SMALL	R		ESSENTIALLY ZERO
TOPLIM	I		UPPER WITHDRAWAL LIMIT
V	R	(100)	RELATIVE VELOCITY PROFILE FOR ONE PORT
VMAX	R		RELATIVE MAXIMUM VELOCITY
WRDEN	R		DENSITY AT WEIR CREST
WRHGT	R		WEIR CREST ELEVATION
WRLNG	R		WEIR LENGTH
WRTYPE	C		"SUBM"ERGED OR "FREE"
X1	R		ELEVATION OF A SEARCH LIMIT
X2	R		ELEVATION OF A SEARCH LIMIT
X3	R		ELEVATION OF A NEW SEARCH LIMIT
X4	R		ELEVATION OF AN OLD SEARCH LIMIT
XFREE	C		"FREE" WEIR
XSUBM	C		"SUBM"ERGED WEIR
XVMAX	R		DISTANCE FROM BOTTOM TO ELEVATION OF MAXIMUM VELOCITY
Y	R		DISTANCE FROM LOCATION OF MAXIMUM VELOCITY TO LOCAL ELEVATION

SUBROUTINE VWEIR

VARIABLE	TYPE	DIMENSION	DESCRIPTION
YLOW	R		DISTANCE FROM LOCATION OF MAXIMUM VELOCITY TO LOWER WITHDRAWAL LIMIT
YTOP	R		DISTANCE FROM LOCATION OF MAXIMUM VELOCITY TO UPPER WITHDRAWAL LIMIT
YVMAX	R		LOCATION OF MAXIMUM VELOCITY REFERENCE TO LOWER WITHDRAWAL LIMIT
ZLOW	R		DISTANCE BETWEEN PORT AND LOWER WITHDRAWAL LIMIT
ZONE	R		DISTANCE FROM LOWER WITHDRAWAL LIMIT TO UPPER WITHDRAWAL LIMIT

SUBROUTINE XFIRST

VARIABLE	TYPE	DIMENSION	DESCRIPTION
BETA	R		PERCENTAGE OF SHORT WAVE RADIATION RETAINED IN TOP TWO FEET OF POOL
BOTTOM	R		ELEVATION OF BOTTOM OF RESERVOIR, FT
DECAY	R		COEFFICIENT FOR EXPONENTIAL DECAY OF EDDY DIFFUSION
DELZ	R		THICKNESS OF A VERTICAL LAYER, FT
DEPTH	R		DEPTH OF POOL, FT
FACTOR	R		CONVERSION FROM K-ACRE FT TO CFS
FGAREA	R		AREA OF FLOOD GATE, SQ FT
FGHGT	R		HEIGHT FROM BOTTOM OF FLOOD GATE CENTERLINE, FT
FGMAX	R		MAXIMUM FLOOD GATE CAPACITY, K-ACRE FT/DAY
FGMIN	R		MINIMUM FLOOD GATE CAPACITY, K-ACRE FT/DAY
FINISH	I		LAST DAY OF SIMULATION
FIRST	I		FIRST DAY OF DATA INPUT
FMAX	R	(8)	MAXIMUM FLOW THRU PORT, K-ACRE FT/DAY
FMIN	R	(8)	MINIMUM FLOW THRU PORT, K-ACRE FT/DAY
FXMAX	R		MAXIMUM FLOOD GATE CAPACITY, CFS
FXMIN	R		MINIMUM FLOOD GATE CAPACITY, CFS
GAMMA	R		ENTRAINMENT COEFFICIENT - PERCENTAGE OF INFLOW VOLUME ENTRAINED FROM SURFACE
K	I		INDEX FOR PORTS
LAMBDA	R		LIGHT EXTINCTION COEFFICIENT - SHAPING COEFFICIENT FOR EXPONENTIAL DISTRIBUTION OF SHORT WAVE RADIATION, 1/FT
LAST	I		LAST DAY OF DATA INPUT
LFILE	I		FILE CODE FOR SIMULATION OUTPUT
LPORT	I	(8)	LAYER CONTAINING EACH PORT
MAXLAY	I		MAXIMUM NUMBER OF LAYERS IN RESERVOIR
MIXCOEF	R		MIXING COEFFICIENT FOR EDDY DIFFUSION
NIP	I		NUMBER OF INFLOW POINTS
NPORTS	I		NUMBER OF SELECTIVE WITHDRAWAL PORTS NOT INCLUDING FLOOD GATE

SUBROUTINE XFIRST

VARIABLE	TYPE	DIMENSION	DESCRIPTION
NQUAL	I		NUMBER OF QUALITIES
NWELL	I	(8)	IDENTIFICATION OF WHICH WETWELL CONTAINS A SPECIFIC PORT
PAREA	R	(8)	PORT AREA, SQ FT
PHGT	R	(8)	HEIGHT FROM BOTTOM OF PORT CENTERLINE
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QVERI	L		.TRUE. - VERIFICATION MODE .FALSE. - PREDICTION MODE
SELMAX	R		MAXIMUM CAPACITY FOR SELECTIVE WITHDRAWAL SYSTEM, K-ACRE FT/DAY
START	I		FIRST DAY OF SIMULATION
SXMAX	R		MAXIMUM CAPACITY FOR SELECTIVE WITHDRAWAL SYSTEM, CFS
TITLE	C	(20)	ALPHANUMERIC IDENTIFICATION OF PROJECT
XMAX	R	(8)	MAXIMUM FLOW THROUGH PORT, CFS
XMIN	R	(8)	MINIMUM FLOW THROUGH PORT, CFS
YEAR	I	(2)	YEARS OF METEOROLOGIC DATA AND HYDROLOGIC DATA

SUBROUTINE XPRINT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
AVQUAL	R	(3)	DOWNSTREAM RELEASE QUALITY FOR EACH QUALITY PARAMETER, MG/L
AVTEMP	R		DOWNSTREAM RELEASE TEMPERATURE, DEG-C
BOTTOM	R		BOTTOM ELEVATION, FT MSL
DAY	I		CURRENT SIMULATION DAY
DEPTH	R		DEPTH OF POOL, FT
DPRINT	I	(14)	SPECIFIED DAYS FOR WHICH DETAILED SIMULATION OUTPUT IS DESIRED
ELEV	R		ELEVATION OF THE POOL CORRESPONDING TO DEPTH
FINISH	I		LAST DAY OF SIMULATION
FIRST	I		FIRST DAY OF DATA INPUT
FLOWIN	R	(3)	INFLOW QUANTITY FOR EACH INFLOW POINT
K	I		INDEX FOR PORTS
KOUNT	I		INDEX FOR DAYS OF DETAILED OUTPUT
L	I		INDEX FOR INFLOW POINTS
LFILE	I		FILE FOR SIMULATION OUTPUT
LINES	I		INDEX FOR LINES ON A PAGE
M	I		INDEX FOR PRINT DAYS
MAXLIN	I		MAXIMUM NUMBER OF LINES PER PAGE
N	I		INDEX FOR PRINT DAYS
NIP	I		NUMBER OF INFLOW POINTS
NOPN	I		NUMBER OF OPEN PORTS
NPDAYS	I		NUMBER OF PRINT DAYS
NPDM	I		NUMBER OF PRINT DAYS - 1
OPEN	I		NUMBER OF OPEN PORTS
OUTFLO	R	(366, 8)	OUTFLOW RATE FOR EACH DAY FOR EACH PORT
PHLOW	R	(3)	OUTFLOW THROUGH EACH OPEN PORT
PORT	I	(3)	PORT NUMBER OF EACH OPEN PORT
QFIRST	L		.TRUE. - FIRST CYCLE THRU THIS SECTION OF CODE - USED TO INITIALIZE PARAMETERS
QHEAD	L		.TRUE. - START NEW PAGE FOR CURRENT SIMULATION DAY OUTPUT

SUBROUTINE XPRINT

VARIABLE	TYPE	DIMENSION	DESCRIPTION
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QPDAY	L		.TRUE. - CURRENT SIMULATION DAY IS A DETAILED PRINT DAY
QUALIN	R	(3, 3)	INFLOW QUALITY FOR CURRENT SIMULATION DAY
QVERI	L		.TRUE. - VERIFICATION MODE .FALSE. - PREDICTION MODE
SAVE	R		TEMPORARY VARIABLE USED FOR SORTING PRINT DAYS
START	I		FIRST DAY OF SIMULATION
STORE	I	(14)	TEMPORARY VARIABLE USED FOR SORTING PRINT DAYS
SUMOUT	R		TOTAL OUTFLOW, CFS
TARGET	R		TARGET TEMPERATURE
TEMPIN	R	(3)	INFLOW TEMPERATURE FOR EACH INFLOW POINT
TITLE	C	(20)	ALPHANUMBERIC IDENTIFICATION OF PROJECT
YEAR	I	(2)	YEARS OF METEOROLOGIC DATA AND HYDROLOGIC DATA

SUBROUTINE XREAD

VARIABLE	TYPE	DIMENSION	DESCRIPTION
A	R		AMPLITUDE FOR HARMONIC RELATION A * SIN (B * T + C) + D USED FOR INPUT OF TARGET TEMPERATURES AND INFLOW TEMPERATURES
B	R		FREQUENCY FOR HARMONIC RELATION A * SIN (B * T + C) + D USED FOR INPUT OF TARGET TEMPERATURES AND INFLOW TEMPEARTURES; CONVERSION FROM DAYS TO RADIANS (B = .0172)
BLANK	C		" " BLANK SPACES
BOTTOM	R		ELEVATION OF BOTTOM OF RESERVOIR, FT
C	R		PHASE SHIFT IN RADIANS FOR HARMONIC RELATION A * SIN (B * T + C) + D USED FOR INPUT OF TARGET TEMPERATURES AND INFLOW TEMPERATURES
CHECK	I		FIRST FOUR ALPHANUMERIC CHARACTERS OF A DATA INPUT CARD IMAGE
D	R		MEAN VALUE FOR HARMONIC RELATION A * SIN (B * T + C) + D USED FOR INPUT OF TARGET TEMPERATURES
DCOEF	R		FREE WEIR DISCHARGE COEFFICIENT
DECAY	R		COEFFICIENT FOR EXPONENTIAL DECAY OF EDDY DIFFUSION
DELZ	R		THICKNESS OF A VERTICAL LAYER, FT
DENC	R	(3)	SPECIFIC GRAVITY OF QUALITY CONSTITUENTS
DINIT	R		INITIAL VALUE OF POOL DEPTH, FT
DPRINT	I	(14)	SPECIFIED DAYS FOR WHICH DETAILED SIMULATION OUTPUT IS DESIRED
DSAVE	R		TEMPORARY VARIABLE - SAVES PREVIOUS VALUE OF INITIAL DEPTH WHILE THE PRESENCE OF A NEW VALUE OF INITIAL IS BEING IDENTIFIED
DUMMY	C	(20)	ONE CARD IMAGE OF INPUT DATA FOR ECHO PRINT OF INPUT
FGANG	R		WITHDRAWAL ANGLE OF FLOOD GATE, RADIANS
FGAREA	R		AREA OF FLOOD GATE, SQ FT
FGHGT	R		HEIGHT FROM BOTTOM OF FLOOD GATE CENTERLINE, FT
FGMAX	R		MAXIMUM FLOOD GATE CAPACITY, CFS
FGMIN	R		MINIMUM FLOOD GATE CAPACITY, CFS

SUBROUTINE XREAD

VARIABLE	TYPE	DIMENSION	DESCRIPTION
FINISH	I		LAST DAY OF SIMULATION
FIRST	I		FIRST DAY OF DATA INPUT
FMAX	R	(8)	MAXIMUM FLOW THRU PORT, CFS
FMIN	R	(8)	MINIMUM FLOW THRU PORT, CFS
GAMMA	R		ENTRAINMENT COEFFICIENT - PERCENTAGE OF INFLOW VOLUME ENTRAINED FROM SURFACE
I	I		INDEX FOR LAYERS
IFILE	I		FILE CODE FOR INPUT OF CONTROL PARAMETERS - SET INTERNALLY TO 05
J	I		INDEX FOR QUALITIES
JFILE	I		FILE CODE FOR INPUT OF HYDROMETEOROLOGICAL DATA
K	I		INDEX FOR PORTS
KFILE	I		FILE CODE FOR OUTPUT OF INFORMATION FOR DEBUGGING
LAMBDA	R		LIGHT EXTINCTION COEFFICIENT - SHAPING COEFFICIENT FOR EXPONENTIAL DISTRIBUTION OF SHORT WAVE RADIATION, 1/FT
LAST	I		LAST DAY OF DATA INPUT
LFILE	I		FILE CODE FOR SIMULATION OUTPUT
LINE	I		LINE NUMBER INCLUDED IN ECHO PRINT OF DATA INPUT
M	I		INDEX FOR TYPES OF PLOTS
MAXLAY	I		MAXIMUM NUMBER OF LAYERS IN RESERVOIR
MFILE	I		FILE USING DURING ECHO PRINT OF INPUT
MIXCOEF	R		MIXING COEFFICIENT FOR EDDY DIFFUSION
N	I		INDEX FOR DAYS
NAME	C	(3)	A) INPUT IDENTIFYING NAMES OF QUALITIES BEING SIMULATED, B) INPUT IDENTIFYING NAMES OF TYPES OF PLOTS
NIP	I		NUMBER OF INFLOW POINTS
NPORTS	I		NUMBER OF SELECTIVE WITHDRAWAL PORTS NOT INCLUDING FLOODGATE
NQUAL	I		NUMBER OF QUALITIES
NWELL	I	(8)	IDENTIFICATION OF WHICH WETWELL CONTAINS A SPECIFIC PORT

SUBROUTINE XREAD

VARIABLE	TYPE	DIMENSION	DESCRIPTION
ONE	I		FIRST DAY OF SPECIAL PRINTOUT FOR DEBUGGING
PAREA	R	(8)	PORT AREA, SQ FT
PHGT	R	(8)	HEIGHT FROM BOTTOM OF PORT CENTERLINE
QECHO	L		.TRUE. - ECHO PRINT OF INPUT DATA IS REQUESTED
QFIRST	L		.TRUE. - FIRST CYCLE THRU THIS SECTION OF CODE - USED TO INITIALIZE PARAMETERS
QINIT	R	(3,100)	INITIAL PROFILE OF QUALITIES
QINITC	L		.TRUE. - INITIAL TEMPERATURE PROFILE IS IN UNITS OF DEGREES CELSIUS
QMORE	L		.TRUE. - INITIAL CONDITIONS CHANGE FOR EACH NEW SIMULATION YEAR
QNKWAL	L		.TRUE. - NO QUALITIES ARE BEING SIMULATED
QOUTC	L		.TRUE. - TARGET TEMPERATURES ARE INPUT IN UNITS OF DEGREES CELSIUS
QOXYG	L		.TRUE. - DISSOLVED OXYGEN IS ONE OF THE SIMULATION QUALITIES
QPORT	L		.TRUE. - PORTS ARE PRESENT AS AN OUTLET DEVICE .FALSE. - NO PORTS
QPROF	L		.TRUE. - OUTPUT WILL BE GENERATED TO PLOT IN-LAKE PROFILES ON DETAILED PRINT DAYS
QRELE	L		OUTPUT WILL BE GENERATED TO PLOT RELEASE TEMPERATURES AFTER EACH SIMULATION YEAR
QVERI	L		.TRUE. - VERIFICATION MODE .FALSE. - PREDICTION MODE
QVINCR	L		.TRUE. - RESERVOIR VOLUMES ARE INPUT AS INCREMENTAL VOLUMES FOR EACH LAYER .FALSE. - RESERVOIR VOLUMES ARE INPUT AS CUMULATIVE VOLUMES AT THE TOP OF EACH LAYER
QWEIR	L		.TRUE. - A WEIR IS INCLUDED AS AN OUTPUT DEVICE .FALSE. - A WEIR IS NOT PRESENT
SELMAX	R		MAXIMUM CAPACITY FOR SELECTIVE WITHDRAWAL SYSTEM, CFS
SETV	R	(3)	SETTLING VELOCITY OF QUALITY CONSTITUENTS, FT/DAY

SUBROUTINE XREAD

VARIABLE	TYPE	DIMENSION	DESCRIPTION
START	I		FIRST DAY OF SIMULATION
TARG	R	(366)	TARGET TEMPERATURES
TINIT	R	(100)	INITIAL TEMPERATURE PROFILE
TITLE	C	(20)	ALPHANUMERIC IDENTIFICATION OF PROJECT
TWO	I	(20)	LAST DAY OF SPECIAL PRINTOUT FOR DEBUGGING
TYPE	C		INDICATES SPECIAL TYPE OF INPUT (CUMULATIVE RESERVOIR VOLUME OR SINE CURVE FOR INPUT TEMPERATURE VALUES)
UNITS	C		UNITS OF INPUT DATA
VOLUME	R	(100)	VOLUME OF RESERVOIR LAYERS, K-ACRE FT
WRHGT	R		HEIGHT FROM BOTTOM OF WEIR CREST, FT
WRLNG	R		WEIR LENGTH
WRTYPE	C		TYPE OF WEIR - "FREE" OF "SUBM"ERGED
WTHETA	R	(8)	WITHDRAWAL ANGLE OF THE OUTLETS
XANGL	C		"ANGL"E
XAREA	C		"AREA"
XBOTT	C		"BOTT"OM
XCELS	C		"CELS"IUS
XCOEF	C		"COEF"FICIENT
XDENC	C		"DENC"
XDEPT	C		"DEPT"H
XENTR	C		"ENTR"AINMENT
XEQUI	C		"EQUI"LIBRIUM TEMPERATURE
XFAHR	C		"FAHR"ENHEIT
XFILE	C		"FILE"
XFLOO	C		"FLOO"D
XFREE	C		"FREE"
XHEAT	C		"HEAT"
XHEIG	C		"HEIG"HT
XINCR	C		"INCR"EMENTAL

SUBROUTINE XREAD

VARIABLE	TYPE	DIMENSION	DESCRIPTION
XINFL	C		"INFL"OW
XINTE	C		"INTE"RVAL
XLAYE	C		"LAYE"RS
XMAXI	C		"MAXI"MUM
XMINI	C		"MINI"MUM
XMIXI	C		"MIXI"NG
XPLOT	C		"PLOT"
XPORT	C		"PORT"S
XPRED	C		"PRED"ICTION
XPRIN	C		"PRIN"T
XPROF	C		"PROF"ILES
XQUAL	C		"QUAL"ITIES
XRELE	C		"RELE"ASE PLOTS
XSELM	C		"SELM"AX
XSETV	C		"SETV"
XSIMU	C		"SIMU"LATION DAYS
XSINC	C		"SINC"URVE
XSTOP	C		"STOP"
XSUBM	C		"SUBM"ERGED
XTARG	C		"TARG"ET TEMPERATURES
XTEMP	C		"TEMP"ERATURE
XTHIC	C		"THIC"KNESS
XVERI	C		"VERI"FICATION
XVOLU	C		"VOLU"MES
XWEIR	C		"WEIR"
XWETW	C		"WETW"ELL
XWIDT	C		"WIDT"H
XYZ	R		TEMPORARY VARIABLE USED IN ESTABLISHING INITIAL QUALITY PROFILES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
August 1993

3. REPORT TYPE AND DATES COVERED
Report 2 of a series

4. TITLE AND SUBTITLE
WESTEX: A Numerical, One-Dimensional Reservoir Thermal Model; Report 2, Programmer's Manual, Version 3.0

5. FUNDING NUMBERS

6. AUTHOR(S)
Darrell G. Fontane, Stacy E. Howington, Michael L. Schneider, Steven C. Wilhelms, editors

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Civil Engineering, Colorado State University, Fort Collins, CO 80523
USAE Waterways Experiment Station, Hydraulics Laboratory, 3909 Halls Ferry Road, Vicksburg, MS 39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army Corps of Engineers, Washington, DC 20314-1000
USAE Waterways Experiment Station, Environmental Laboratory, 3909 Halls Ferry Road, Vicksburg, MS 39180-6199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
Instruction Report W-93-2

11. SUPPLEMENTARY NOTES
Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

WESTEX is a numerical, one-dimensional reservoir thermal model developed at the US Army Engineer Waterways Experiment Station. This report was developed to assist users of the WESTEX computer model. The report provides the listing of the WESTEX FORTRAN code and a list of varible definitions, and serves as supplementary information for Report 1 of this two-part series documenting the WESTEX model.

14. SUBJECT TERMS
Reservoir operation
Reservoir release quality
Selective withdrawal
Stratification
Thermal modeling
Water quality modeling

15. NUMBER OF PAGES
136

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT